

UNIVERZITA KOMENSKÉHO BRATISLAVA
Fakulta matematiky, fyziky a informatiky

Simulácia impulzných neurónových
sietí pomocou udalostí, model JASTAP

Martin Svetlák
Bratislava, Máj 2007

Simulácia impulzných neurónových sietí pomocou udalostí, model JASTAP

Diplomová práca

Martin Svetlík

UNIVERZITA KOMENSKÉHO BRATISLAVA
Fakulta matematiky, fyziky a informatiky

Študijný program: Informatika

Ing. Igor Farkaš, PhD.
Nuno C. Marques, MSc., PhD.

Bratislava, Máj 2007

PREHLÁSENIE

Prehlasujem, že prácu som vypracoval samostatne pod vedením vedúcich diplomovej práce a nečerpал som z inej, ako uvedenej literatúry.

.....

POĎAKOVANIE

Úprimné poďakovanie patrí vedúcemu mojej diplomovej práce na Slovensku, Ing. Igorovi Farkašovi, PhD. za rady a ochotu pri konzultáciách. Nemenej by som rád poďakoval Nuno C. Marquesovi, MSc., PhD. za obetovaný čas, ktorý mojej práci venoval počas štúdia v Portugalsku a za prínosné rady, ktoré mi pomohli vydať sa správnym smerom. Takisto poďakovanie patrí mojej rodine a priateľom za podporu počas celého štúdia.

PREDSLOV

Tajomstvo fungovania mozgu fascinuje vedcov už stáročia. Pokúšajú sa pochopiť mechaniku jeho fungovania zo všetkých perspektív. Jedným z pohľadov na fungovanie mozgu je skúmanie fungovania elementárnych jednotiek, z ktorých je vytvorený. Preto sa časť ľudí začala zaoberať fungovaním neurónov. Experimentálne pokusy viedli k porozumeniu zákonitostí, ktorými sa neurón riadi a mnohé teoretické modely sa snažili popísať tieto pravidlá.

Priebehom minulého storočia vznikali mnohé matematické a logické modely, ktoré popisovali určité aspekty správania neurónov. Ale ako veda postupuje ďalej, vznikajú podrobnejšie modely a komplikovanejšie mechaniky ako simulovať správanie týchto modelov. Najväčšia pozornosť sa v poslednom čase venuje tzv. impulzným modelom neurónov, ktoré sú tými najdetailnejšími modelmi neurónu.

Ako však pristupovať k simulácii neurónov pomocou impulzných modelov na terajších počítačoch? Aj tejto otázke sa budem v práci venovať.

Martin Svetlík
Bratislava, Máj 2007

ABSTRAKT

V práci sa budem venovať impulzným modelom neurónu všeobecne. Spravím prehľad základných typov modelov a ich vzájomných vzťahov. Zamyslím sa nad teóriami neurálneho kódu, ktoré sa predpokladajú v jednotlivých modeloch a neskôr navrhmem zaujímavú možnosť ako simulovať samotné impulzy. Zameriam sa na výsledky dosiahnuté pomocou Izhikevichovho modelu a niektoré vlastnosti tohto modelu sa pokúsím aplikovať na model JASTAP. V hlavnej časti práce sa budem venovať navrhnutým úpravám modelu JASTAP pre účel simulovania jeho správania pomocou udalostí, čo je prirodzený spôsob simulovania impulzov, ak berieme ohľad na presnosť ich časovania. Ukážem problémy spojené s týmto druhom simulácie modelu JASTAP ale aj výhody plynúce z tohto prístupu k simulácii.

V druhej časti práce aplikujem navrhnutý výpočtový model na problém riešiteľný impulznými modelmi neurónov v praxi. A načrtnem postavenie mnou navrhnutého systému v porovnaní s terajšími spôsobmi simulácie. Ku koncu navrhmem nové oblasti v ktorých by sa tento model mohol úspešne uplatniť.

Kľúčové slová: Impulzné modely neurónov, Impulz, Udalosť, JASTAP

Obsah

1	Úvod	11
1.1	Úvod a motivácia	11
1.2	Prehľad	12
2	Impulzné modely neurónov	13
2.1	Neurón	13
2.2	Vznik impulzných modelov neurónu	15
2.3	Kódovanie informácie impulzmi	16
2.4	Detailné modely	18
2.5	Formálne impulzné modely neurónov	18
2.5.1	Neurón ako Integrátor	19
2.5.2	Model SRM	20
2.5.3	Model SRM ₀	22
2.5.4	Izhikevichov model	23
2.5.5	Synchronizácia a polychronizácia	25
3	Model JASTAP a jeho modifikácie	29
3.1	Pôvodný model JASTAP	29
3.1.1	Parametre a vlastnosti modelu	29
3.1.2	Simulácia	31
3.2	Navrhnuté modifikácie	32
3.3	Simulácia v reálnom čase	33
3.3.1	Manažovanie udalostí	35
3.3.2	Výpočet membránového potenciálu	37
3.3.3	Výpočet času vzniku impulzu	40
3.3.4	Aproximačný iteračný algoritmus	44
3.4	Možné vylepšenia	47
3.5	Testy efektívnosti	49
3.6	Paralelizácia	52
4	Samoorganizujúca mapa pomocou JASTAP	55
4.1	Impulzné samoorganizujúce mapy	55

4.2	Samoorganizácia	55
4.3	IRIS	57
4.4	Sieť	58
4.4.1	Laterálne spojenia	58
4.4.2	Normalizácia vstupu	60
4.4.3	Priebeh učenia	61
4.4.4	Záver	65
5	Záver a budúci výskum	67
A	Príloha: Iris	68

Zoznam obrázkov

1	Neurón	13
2	Obvod neurónu ako integrátora.	19
3	Graf MP a pod ním $I(t)$	20
4	Graf možného priebehu hodnoty MP v modeli SRM	22
5	Bohatosť správání Izhikevichovho modelu pri správnom nastavení parametrov	25
6	Polychronizácia	26
7	STDP krivka	28
8	Funkcia PSP a jej derivácia	38
9	Graf funkcie mexického klobúka	59
10	Učenie	62
11	Skupiny	64

1 Úvod

1.1 Úvod a motivácia

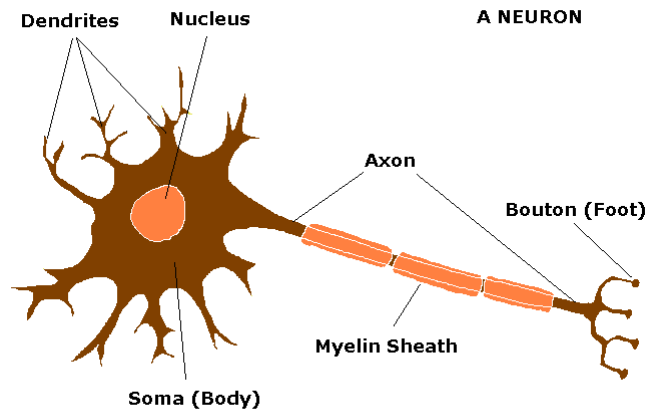
Popisu fungovania a modelovania neurónov sa venujú neurovedci a informatici už stovku rokov. Modely neurónov sa počas týchto rokov vyvíjali. Počnúc základnými modelmi o ktorých bolo dokázané že môžu simulovať boolovské funkcie sa modely postupne komplikovali. Až sa prišlo na úroveň kde model podrobne popisoval neurón, jeho vnútornú dynamiku a jeho reakcie na podnety z okolia. S vývojom týchto komplexných modelov začala byť zaujímavá otázka: Ako vlastne prebieha prenos informácie v mozgu, ako je tá informácia kódovaná?. Je známe, že prenos informácie je realizovaný pomocou elektrických impulzov medzi neurónmi. Ale v mozgu je mnoho iných mechanizmov ktoré vplývajú na správanie neurónov. Ako hladiny neurotransmiterov, množstvá rôznych iónov a ďalších látok v mozgu. Aj vďaka tomu že informačný tok v mozgu je veľmi komplexný vzniklo veľa teórií, hovoriacich o tom ako môžu neuróny komunikovať. Tie jednoduchšie boli priamo pozorované v živých neurónoch napr. pri pohybe svalov. A zároveň bolo dokázané, že určite existujú iné, oveľa komplexnejšie mechanizmy kódovania informácie. Niektoré priamo pozorované v živých tkanivách, iné odvodené teoreticky. Čím ďalej sa táto oblasť skúma, tým viac z modelov komunikácie (a kódovania) je pozorovaných v živých systémoch. Napriek tomu je veľká časť fungovania mozgu stále záhadou.

V posledných rokoch začína prevládať názor, že presné časovania jednotlivých impulzov majú veľký význam pre prenos informácie. Taktiež veľký dôraz sa začína klásť na synchronizáciu skupín a to v spojení s Izichevicovou teóriou polychronizácie čo kladie nároky na presnosť simulačného prístupu. V práci sa teda zamýšľam nad možnosťou simulácie impulzného modelu JASTAP pomocou udalostí a nie v časových krokoch ako pracuje mnoho iných modelov. Čo umožňuje prirodzeným spôsobom simulovať impulzy so skoro neobmedzenou presnosťou. Ako v práci ukážem, tento spôsob je aj vhodný na paralelizáciu, čiže veľké a biologicky relevantné siete ním môžu byť jednoducho simulované. V práci uvádzam možnosti ako aj výhody a nevýhody tohto prístupu. Nakoniec otestujem tento spôsob simulácie na reálnej aplikácii.

1.2 Prehľad

- **1.** V prvej kapitole sa úvodom venujem aktuálnym impulzným modelom neurónov a porovnávam ich. Uvediem čím som sa v ich mechanike inšpiroval a ako sa dajú myšlienky v nich použiť.
- **2.** V druhej kapitole sa priamo venujem modelu JASTAP. Rozoberiem mechaniku tohto modelu a navrhnem zmeny potrebné pre simuláciu pomocou udalostí. Popíšem samotný mechanizmus simulácie a načrtnem jeho výhody, nevýhody a možnosti.
- **3.** V poslednej kapitole aplikujem tento výpočtový mechanizmus na reálny problém, kde bude vidieť jeho kladné aj záporné vlastnosti v porovnaní s klasickým prístupom.

Obrázok 1: Neurón



2 Impulzné modely neurónov

2.1 Neurón

Základnou výpočtovou jednotkou v mozgu je neurón. V ľudskom mozgu môžeme v kôre nájsť až 10^4 neurónov na 1mm^2 a kilometre spojení medzi nimi. Pritom aké je neurónov ohromné množstvo, každý z nich je aj komplexný výpočtový mechanizmus. Preto ich možnosti a obmedzenia ľudí odjakživa fascinovali. K tomu všetkému môžeme ešte pridať fakt, že poznáme veľa druhov neurónov, ktoré sa dosť zásadne líšia svojim správaním a dynamikou. Každý z týchto druhov je zodpovedný za niečo iné a sú špecializované na plnenie svojej úlohy.

Pozrime sa na neurón teraz podrobnejšie. Bunka sa skladá z tela, ktoré obsahuje jadro. Ďalej z dendritov, ktoré sa stromovito rozvetvujú a sú pokryté synaptickými spojeniami slúžiacimi hlavne na prijímanie impulzov od iných neurónov. Každý neurón má práve jeden axón. Ten slúži ako výstup, šíri elektrickú vlnu (impulz) ďalej, často do úplne inej časti mozgu alebo inej časti tela živočicha. Axón je pokrytý myelínovými obalmi, ktoré zabezpečujú uchovanie intenzity signálu pri prenose na väčšiu vzdialenosť. Na konci je rozvetvený a pomocou synaptických spojení prenáša impulz z pre-synaptického na post-synaptický neurón. Prenos prebieha pomocou neurotransmitera, ktorý je uvoľnený pri dorazení impulzu na pre-synaptickú časť synaptického spojenia a spôsobí otvorenie

iónových kanálov na membráne post-synaptického neurónu. Zmenu napätia v neuróne spôsobenú otvorením týchto kanálov nazývame post-synaptický potenciál.

Na popis dynamiky neurónu popisujeme používame viaceré pojmy. Tieto pojmy sú spoločné pre prevažnú väčšinu modelov neurónu. Popíšme si ich.

- **Membránový potenciál** - (ďalej iba MP). Vyjadruje elektrické napätie medzi vnútrom bunky a okolitým prostredím. Napätie vzniká rôznymi koncentraciami iónov (Na^+ , K^+) na opačných stranách membrány. Membrána obsahuje iónové kanály, ktoré sa môžu otvárať a prepúšťať ióny na druhú stranu meniac tak napätie na membráne. Pokojová hodnota MP sa pohybuje okolo $-65mV$. K tejto hodnote sa MP blíži vždy, keď nie je dlhšiu dobu vychýlený impulzom.
- **Post-synaptický potenciál** - Keď dorazí impulz do neurónu cez nejaký synaptický spoj, spôsobí zmenu MP. Túto zmenu voláme post-synaptický potenciál (ďalej iba PSP). V závislosti na synaptickom spojení môže impulz spôsobiť depolarizáciu neurónu ak ide o excitačné spojenie. Tu dôjde k excitačnému PSP (ďalej iba EPSP) a MP sa na čas posunie z kladnejším hodnotám. Ak je spojenie inhibičné, tak dôjde k hyperpolarizácii. Tu hodnota MP klesne ešte viac k záporným hodnotám, čiže PSP je inhibičný (IPSP).
- **Aktivačný prah** - Je to prahová hodnota MP, pri ktorej sa v tele bunky vygeneruje impulz. Tento jav nazývame akčný potenciál. Hodnota tohto prahu sa pohybuje približne $20 - 30mV$ nad pokojovou hodnotou. Aktivačný prah sa implementuje ako pevná hranica alebo jemne kolísajúca hodnota.
- **Akčný potenciál** - (ďalej iba AP) Ide o rapídnu zmenu hodnoty MP, ktorá nastane keď MP prekročí aktivačný prah. Keď MP prekročí prah, dôjde k silnej vlne polarizácie s amplitúdou okolo $100mV$. Táto vlna sa ako impulz presunie cez axón ďalej do ďalších neurónov a v nich spôsobí PSP. Po tejto vlne sa membránový potenciál znova hyperpolarizuje, brániac tým aby v najbližšom čase znova došlo ku

vygenerovaniu impulzu.

- **Absolútna refraktórna fáza** (Absolute Refractory period)- Časový interval po vyslaní impulzu, počas ktorého nie je neurón schopný vyslať ďalší impulz. Je spôsobená hyperpolarizáciou MP po vyslaní impulzu.
- **Oneskorenie spojenia** - Každý axón pri prenose impulzu spôsobí nejaké oneskorenie, to jest čas za ktorý sa impulz axómom šíri + čas ktorý potrebuje synaptické spojenie na prenos impulzu. Sčítaním týchto dvoch časov získame rozdiel medzi dobou generovania impulzu a začiatkom PSP spôsobeného týmto impulzom v post-synaptickom neuróne. Tento čas označíme ako oneskorenie spojenia. Málokteré modely berú toto oneskorenie do úvahy (najmä rozdiely v oneskorení pre rôzne spojenia). Ja ale budem toto oneskorenie simulovať presne.

2.2 Vznik impulzných modelov neurónu

Umelé neurónové siete sú už teraz starou technikou, ktorej prvé idey a modely sa objavili pred viac ako 50 rokmi. Prvá generácia modelov začala výtvorom McCullocha a Pittsa. Ich binárny model bol prvý, ktorý bol úspešne použitý v mnohých aplikáciách. Pomocou nich sa dala zostrojiť ľubovoľná binárna funkcia za pomoci viacerých vrstiev týchto neurónov. Ďalšou generáciou modelov boli tie so spojitou aktivačnou funkciou, ich výstupom už nebol binárny signál ale určitá reálna intenzita. Táto intenzita predstavovala strednú hodnotu frekvencie impulzov vysielaných z tohto neurónu. Pre siete zložené z takýchto neurónov bolo navrhnutých veľa učiacich algoritmov (napr. Back-Propagation), ktoré umožnili učenie viacvrstvových sietí. Tieto siete sa tým stali univerzálnym aproximátorom, schopné aproximovať ľubovoľnú funkciu definovanú na konečnom počte bodov. Táto druhá generácia je zatiaľ určite tá najpoužívanejšia vďaka širokej palete modelov a ich využití. Posledné roky sa však do popredia tiež dostáva tretia generácia modelov, tzv. impulzné modely neurónov. Ich zameraním už nie je stredná hodnota počtu impulzov na neurálnej dráhe za časový interval ale jednotlivé impulzy a ich presné časovanie. Tieto modely sa podrobnejšie venujú aj vnútornej dynamike jednotlivých neurónov, čo z nich robí biologicky najpodrob-

nejšie modely. Posledné vedecké práce ukazujú väčšiu výpočtovú silu aká je očakávateľná od modelov druhej generácie. Na určité úlohy ich stačí menej a sú schopné zachytiť aj správanie ktorého staršie modely nie sú schopné. Ich opodstatnenie samozrejme posilňujú aj výsledky hovoriace o tom, že mnohé prenosy informácií v mozgu nemôžu prebiehať pomocou frekvencie impulzov, ale iba presným časovaním jednotlivých impulzov. Viac o tom už však v ďalšom odseku. Pre viac detailov porovnávajúcich jednotlivé triedy modelov čitateľ môže nahliadnuť do [1] alebo [19].

2.3 Kódovanie informácie impulzmi

V mozgu cicavca je viac ako 10^{10} neurónov, ktoré sú spolu pospájané a navzájom komunikujú. V každej aj malej časti mozgu sú každú milisekundu vyslané tisícky impulzov. Závažná otázka ktorú si neurovedci, psychológovia a informatici kladú je: Aká informácia je zakódovaná v týchto impulzoch? Aký kód používajú neuróny na zakódovanie informácie ktorú posielajú ďalej? Ako post-synaptické neuróny dekodujú túto informáciu? Môžeme my, ako vonkajší pozorovatelia vyčítať a pochopiť informáciu prenášanú impulzmi?

Tieto otázky zatiaľ však nemajú jasnú odpoveď. Prvým mechanizmom kódovania, ktorý sa v neurónoch v minulosti očakával bolo kódovanie frekvenciou (Rate Coding). Táto teória predpokladala, že neuróny posielajú určitú hodnotu zakódovanú vo frekvencii (strednej hodnote frekvencie) impulzov. Teória bola podložená mnohými experimentálnymi pokusmi za posledných 80 rokov, keď už v 30. rokoch minulého storočia bolo ukázané, že intenzita impulzov v svalových receptoroch je priamo úmerná sile ktorú je svalom vytvorená. Posledných 30 rokov však táto teória začína byť kritizovaná kvôli neschopnosti popísať niektoré druhy správania neurónov. V predchádzajúcich rokoch stále viac experimentov ukazuje, že existujú procesy v mozgu, v ktorých určite tento spôsob kódovania neprebíha.

Napríklad pokusy kde bolo ukázané, že človek dokáže rozpoznať obraz a reagovať naň stlačením tlačidla v čase pod 400 ms. Ďalej bolo ukázané, že človek dokáže spozorovať a identifikovať obrázok v sérii mnohých iných obrázkov ak mu bol každý z nich ukázaný iba 100 ms. Alebo experimentami podložený fakt, že mucha dokáže reagovať na vonkajší podnet a zmeniť smer letu za menej ako 40 ms. Tieto experimentálne výsledky ukazujú,

že neuróny jednoducho nemajú vždy čas odhadovať priemernú frekvenciu impulzov. A pri šume, ktorý je v mozgu prítomný by každý neurón potreboval dlhé časové okno na spoľahlivý odhad reálnej frekvencie vysielanej pre-synaptickým neurónom. Preto v mnohých častiach mozgu musí existovať aj iný mechanizmus, ktorý dokáže spracovávať informáciu rýchlejšie.

Aj kvôli týmto výsledkom spolu so vznikom impulzných modelov neurónov začali vznikať iné teórie popisujúce prenos a kódovanie informácie medzi neurónmi. Tie boli založené na novom predpoklade, že informáciu neprenáša iba intenzita prúdu impulzov, ale aj časovanie jednotlivých akčných potenciálov (impulzov). Vzniklo niekoľko všeobecných prístupov popisujúcich ako môže toto presné časovanie preniesť informáciu:

- **Čas prvého impulzu** (Time-to-First-Spike) Tento prístup vychádza z pôvodných myšlienok kódovania frekvenciou. Zjednodušene predpokladá, že čím väčšiu frekvenciu impulzov chce neurón vyslať ako reakciu na podnet, tým skôr dorazí do post-synaptického neurónu prvý z nich. Táto predstava síce vysvetľuje ako by bolo možné spracovávať informáciu rýchlejšie, je však značne zjednodušená a naráža v praxi na veľa problémov hlavne ohľadom šumu, ktorý je v mozgu prítomný.
- **Fáza** (Phase) V mnohých častiach mozgu je pozorovateľná určitá globálna oscilácia aktivity v pozadí. Keď je veľká časť neurónov aktivovaná pravidelne v nízkofrekvenčných cykloch. Jedna z možností kódovania informácie presným časovaním impulzu je, že informácia je uložená v časovom posune tohto impulzu vzhľadom na osciláciu pozadia (Background Oscillation). Táto možnosť bola aj podložená experimentálne.
- **Korelácia a synchronizácia** (Correlation and Synchrony) Táto posledná teória je založená na možnosti, že význam majú impulzy vo vzájomnom spojení. Že samotná informácia je v synchronizácii impulzov od rôznych pre-synaptických neurónov. Známa je myšlienka, že zosynchronizovanie viacerých impulzov by mohlo znamenať, že patria spolu.

Je ešte veľa rôznych iných teórií popisujúcich prípadnú možnosť prenosu informácie, či už intenzitou signálu alebo presným časovaním jednotlivých impulzov a určite mnohé z

nich sú v niektorej časti neurálnej sústavy realizované. Čitateľ sa môže o týchto konceptoch dozvedieť viac v [1]. Ja v mojej práci navrhujem model, ktorý predpokladá, že hlavná časť informácie je uložená v presnom časovaní impulzov. Najviac ma však zaujala teória o synchronizácii impulzov, ktorou som bol značne inšpirovaný.

2.4 Detailné modely

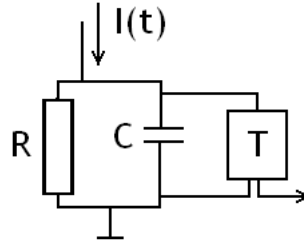
Najklasickejším príkladom detailného modelu je model Hodgkina a Huxleyho (Hodgkin-Huxley). Jedná sa o jeden z najkomplexnejších modelov simulujúcich dynamiku neurónu až na úroveň jednotlivých druhov iónových kanálov. Jeho autormi sú páni Hodgkin a Huxley, ktorí skúmali v roku 1952 gigantický neurón kalamárov (Squid). Výsledkom ich práce bol komplexný model simulujúci dynamiku neurónu pomocou tokov rôznych iónov v jednotlivých iónových kanáloch. Vznikol tým 4-dimenzionálny model popísaný diferenciálnymi rovnicami, ktoré boli odpozorované zo živých neurónov, ktoré skúmali. Jedná sa o známy model, ktorý bol základom pre mnoho iných modelov. Tie vznikali zjednodušením modelu Hodgkina a Huxleyho redukciou zo 4 dimenzií na 2 dimenzie. Proces redukcie zjednodušuje koncept viacerých druhov iónových kanálov a je aplikovateľný aj na iné detailné modely neurónov počítajúce s viacerými druhmi kanálov a iónov. Ak čitateľa zaujíma viac podrobností, odporúčam pozrieť [1].

2.5 Formálne impulzné modely neurónov

Na to aby sme boli schopní reálne simulovať správanie väčšej neurónovej siete, musíme abstrahovať od mnohých aspektov neurálnej dynamiky. Musíme opísať správanie jedného neurónu jednoduchšie, ale nezanedbať dôležité vlastnosti natoľko, aby sme zachovali pôvodné vzorce správania. To je miesto, kde sa treba zamyslieť ktoré mechanizmy umožňujú neurónu splňať jeho úlohu, spracovávanie informácie, a ktoré môžeme zanedbať.

Hlavný predpoklad týchto modelov je, že všetky impulzy sú uniformné. Čiže samotné neprenášajú žiadnu informáciu a sú vždy identické. Preto sa v týchto modeloch predpokladá, že samotná informácia je kódovaná v časovaní impulzov. Vo všeobecnosti sa budeme najmä venovať modelom založeným na premennej u (hodnota membránového potenciálu). Správanie týchto modelov sa odvíja od vstupov a od hodnoty tejto premennej. Zároveň

Obrázok 2: Obvod neurónu ako integrátora.



reakcia hodnoty u na vstupy závisí na danom modele a dynamike ktorú popisuje.

2.5.1 Neurón ako Integrátor

Model integrátora (Integrate-and-fire model) sa na neurón pozerá ako na elektrický obvod. Interpretuje neurón ako kondenzátor C paralelne zapojený s rezistorom R podporovaný vstupným prúdom $I(t)$. Ak napätie na kondenzátore prekročí hodnotu T (aktivačný prah), tak dôjde k vzniku impulzu a napätie na kondenzátora sa vybije (viď. Obr 2.).

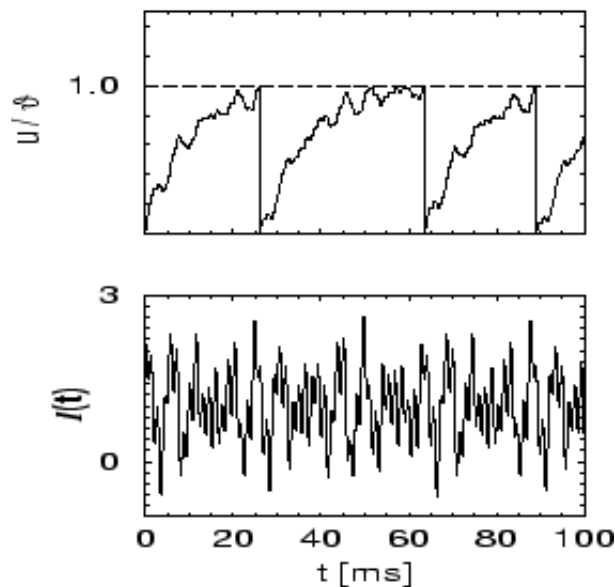
Celý mechanizmus sa dá popísať diferenciálnou rovnicou a podmienkou. Pre podrobnosti (viď: [1])

$$\tau_m \frac{du}{dt} = -u(t) + RI(t)$$

$$\mathbf{IF} \quad u(t) = \vartheta \quad \mathbf{THEN} \quad u(t) := u_r$$

Z diferenciálnej rovnice vieme vyčítať, že čím väčší je membránový potenciál, tým rýchlejšie aj časom klesá. A jeho hodnota je značne riadená hodnotou $I(t)$ vstupného prúdu. Keďže funkcia času $I(t)$ môže byť ľubovoľná funkcia, tak môžeme simulovať model nie len na prichádzajúcich impulzoch, ale na rôznych vstupných prúdoch všeobecne.

Obrázok 3: Graf MP a pod ním $I(t)$.



Môžeme si všimnúť (Obr 3.) ako vyzerá graf membránového potenciálu (horný graf) pri určitom vstupnom prúde (dolný graf). Keď hodnota membránového potenciálu dosiahne hodnotu aktivačného prahu tak vyšleme impulz a jednoducho vynulujeme hodnotu MP na pokojovú hodnotu. Ako dodatočná vlastnosť modelu sa doneho môže pridať ešte absolútnu refraktornú fázu, ktorá funguje tak že sa hodnota MP vynuluje nie len v okamihu vzniku impulzu, ale je vynulovaná na nejaký čas a normálna dynamika sa obnoví až po ukončení tohto časového okna.

Vo všeobecnosti do tejto kategórie patrí veľa modelov, ktoré sa mierne líšia presným znením diferenciálnej rovnice (systémom rovníc).

2.5.2 Model SRM

Na rozdiel od predošlého prístupu, popisu dynamiky pomocou diferenciálnych rovníc, SRM model (Spike Response Model) funguje trochu jednoduchšie. Tento model (trieda modelov) je síce tiež založený na hlavnej premennej u (membránovom potenciáli), ale tu sa hodnota

MP počíta z rôznych udalostí, ktoré sa udiali v minulosti daného neurónu. Teda nie je popísaný diferenciálnymi rovnicami. Hodnota MP v čase t je určitá sumarizácia všetkého čo sa udialo za určité relevantné časové okno. Vo všeobecnosti vyzerá vzťah na výpočet MP v čase t takto:

$$u(t) = \eta(t - t') + \sum_i w_j \sum_f \epsilon(t - t', t - t_j^{(f)}) + \int_0^\infty \kappa(t - t', s) I^{ext}(t - s) ds$$

Aby sme tomuto vzťahu lepšie porozumeli, popíšeme si jeho jednotlivé časti. Začnime s premennými. Premenná t vyjadruje čas v ktorom chceme vedieť hodnotu membránového potenciálu. Ďalej t' je čas posledného impulzu, ktorý daný neurón vygeneroval. $t_j^{(f)}$ vyjadruje čas, kedy daný neurón dostal impulz od j -teho pre-synaptického neurónu. Hodnota w_j vyjadruje silu synaptického spojenia medzi daným neurónom a j -tym pre-synaptickým neurónom.

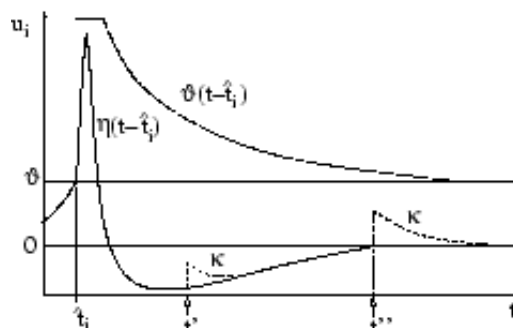
Ďalej si popíšeme jednotlivé funkcie:

Funkcia $\eta(t - t')$ popisuje reakciu neurónu na impulz, ktorý v minulosti ako posledný vyslal. Čiže v podstate predstavuje vplyv akčného potenciálu na MP. Vidieť že hodnota funkcie závisí iba od toho ako dávno bol impulz vyslaný ($t - t'$).

Funkcia $\epsilon(t - t', t - t_j^{(f)})$ predstavuje reakciu MP na impulzy z pre-synaptických neurónov. Čiže ide o post-synaptický potenciál. Ako si môžeme všimnúť, hodnota funkcie závisí od toho ako dávno bol prijatý impulz z pre-synaptického neurónu ($t - t_j^{(f)}$). Ale zároveň aj čas, kedy naposledy daný neurón vyslal svoj vlastný impulz ($t - t'$) vplýva na hodnotu funkcie. Z toho vyplýva že priebeh PSP bude rôzny v závislosti od vnútorného stavu neurónu.

Funkcia $\kappa(t - t', s)$ zase vyjadruje reakciu membránového potenciálu na rôzne vstupné prúdy, ktoré ovplyvňovali stav neurónu v minulosti. Táto funkcia tiež vyjadruje, že neurón inak reaguje na vstupujúce prúdy priamo po vygenerovaní impulzu a dlhšie po ňom. Ide

Obrázok 4: Graf možného priebehu hodnoty MP v modeli SRM



tu o snahu simulovať zmeny v priepustnosti membrány neurónu pri vzniku impulzu.

Vstup $I(t - s)$ vyjadruje intenzitu vstupného prúdu do neurónu v danom čase.

Na (obr 4) vidíme ako by mohol vyzeráť priebeh hodnoty membránového potenciálu SRM modelu. Po vygenerovaní impulzu dôjde vďaka funkcii $\eta()$ k hyperpolarizácii a hodnota MP rapídne klesne. Ďalej si môžeme všimnúť že impulz, ktorý dorazil do neurónu tesne po akčnom potenciáli, mal oveľa menší vplyv na membránový potenciál ako ten ktorý dorazil neskôr. Toto je tiež spôsobené tým, že model sa snaží simulovať zmeny v priepustnosti membrány.

Čo sa týka absolútnej refraktornej fázy, v tomto modeli existuje niekoľko spôsobov ako danú vlastnosť nasimulovať. (I) Navrhnuť funkcie tak, aby po vyslaní impulzu neurón nereagoval skoro vôbec ani na impulzy ani na vstupný tok. (II) Dočasne zvýšiť hodnotu aktivačného prahu, tak aby bolo nemožné počas potrebnej doby prah dosiahnuť. (III) Navrhnuť funkciu akčného potenciálu tak, aby spôsobila značnú hyperpolarizáciu.

2.5.3 Model SRM_0

Pre rôzne účely nám môže stále aj SRM model pripadať moc komplexný. Preto zjednodušením jeho základného vzťahu môžeme dostať SRM_0 model. Ten všetky predtým spomenuté funkcie ($\epsilon()$, $\kappa()$) zjednodušuje a ignoruje prvý parameter. Cieľom tohto zjednodušenia je spraviť tvar funkcie post-synaptického potenciálu nezávislým od stavu neurónu.

Taktiež jeho reakciu na vonkajšie vstupné prúdy spraviť nezávislú a celý čas rovnakú.

$$\epsilon_0(s) = \epsilon(\infty, s)$$

$$\kappa_0(s) = \kappa(\infty, s)$$

Po týchto zmenách sa obe funkcie správajú, ako keby daný neurón nikdy žiadny impulz nevyslal a nikdy v ňom k akčnému potenciálu nedošlo. Výsledná rovnica nakoniec vyzerá takto:

$$u(t) = \eta(t - t') + \sum_i w_j \sum_f \epsilon(t - t_j^{(f)}) + \int_0^\infty \kappa(s) I^{ext}(t - s) ds$$

Tento model nám umožňuje použiť rovnakú funkciu na priebeh každého post-synaptického potenciálu. Stačí naškálovať amplitúdu PSP podľa váhy synaptického spojenia w_j . A ďalej pre výpočet MP stačí tieho funkcie správne časovo zosumovať a zistiť, či došlo k prekročeniu aktivačného prahu. A ak došlo k prekročeniu, tak aplikujeme funkciu η a membránový potenciál hyperpolarizujeme.

V tomto modeli sa už blížime ideologicky k modelu JASTAP, na ktorý sa v práci chcem najviac zamerať. Model JASTAP je v podstate SRM_0 model s malým zjednodušením, že ako vstupy berie iba pre-synaptické impulzy a nie vstupné prúdy.

2.5.4 Izhikevichov model

Ďalší model, ktorý bol pre mňa obzvlášť zaujímavý, je Izhikevichov model (viď [2]). Čo je na tomto modeli zaujímavé? Ide o návrh, ktorý je biologicky realistický a pritom stále výpočtovo jednoduchý. Autorovým cieľom bolo navrhnúť mechaniku, ktorá dokáže simulovať pestré správanie biologických neurónov podobne ako detailné modely podobné modelu Hodgkina a Huxleyho. A pritom je jednoduchá a výpočtovo efektívna používajúca prístup neurón ako integrátor. Výsledkom je síce model ktorý je biologicky nerealisticky jednoduchý, ale pri tom stále schopný širokého spektra správání. Čím autor dosiahol model, ktorým je možné simulovať aj veľké siete zložené z neurónov schopných bohatej dynamicky.

Pomocou Bifurkačných metód (viď [3]) je možné realistický a komplexný model Hodg-

kina a Huxleyho zjednodušiť na 2 dimenzie. Dostaneme tým systém dvoch jednoduchých diferenciálnych rovníc, ktoré Izhikevich navrhol nasledovne.

$$v' = 0.04v^2 + 5v + 140 - u + I$$

$$u' = a(bv - u)$$

K tomu ešte podmienka pre vznik akčného potenciálu:

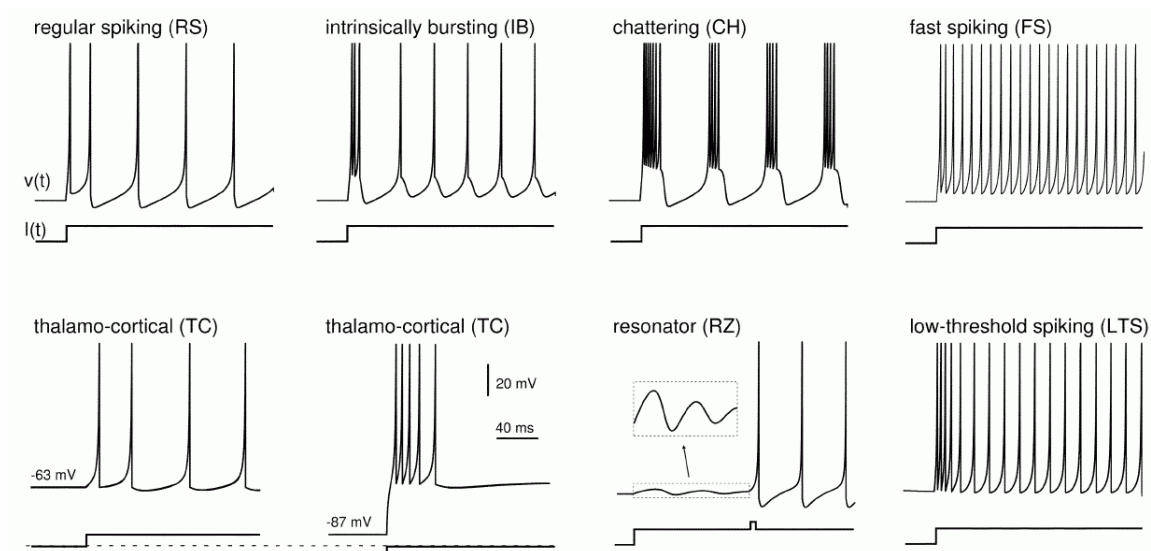
$$\mathbf{IF} \quad u(t) = \vartheta \quad \mathbf{THEN} \quad u(t) := u_r \quad (1)$$

Rovnice obsahujú dve premenné u a v a parametre a , b , c , d . Premenná v reprezentuje membránový potenciál a premenná u zase obnoviteľnosť membrány. Číže u vyjadruje aktiváciu prúdov K^+ a deaktiváciu prúdov Na^+ , čím predstavuje negatívny vplyv pre rast premennej v . Keď hodnota v membránového potenciálu dosiahne $30mV$ tak nastane resetnutie u a v ako vyjadruje podmienka (1). Hodnota I vyjadruje vstupný tok do neurónu, čiže vstup z okolia.

Model bol navrhnutý tak, aby odpovedal dynamikou neurónom mozgovej kôry. Ako vidieť, správanie je nastaviteľné pomocou 4 parametrov. Vďaka nim sú možnosti dynamiky jednotlivých neurónov veľmi bohaté. Popíšme si jednotlivé parametre:

- Parameter a určuje časovú škálu premennej u . Čím je hodnota vyššia tým rýchlejšie sa dokáže membrána obnoviť. Typická hodnota je $a = 0.02$.
- Parameter b predstavuje silu väzby medzi membránovým potenciálom v a zmenou premennej u . Čím je hodnota b vyššia tým je premenná u náchylnejšia na podprahové fluktuácie membránového potenciálu. Typická hodnota je $b = 0.2$.
- Parameter c určuje hodnotu na ktorú resetneme membránový potenciál po vygenerovaní impulzu. To znamená, že určuje tvar akčného potenciálu. Štandardná hodnota je $c = -65mV$.
- Parameter d ovplyvňuje hodnotu premennej u po prekročení prahu a vygenerovaní impulzu. Ako vidieť parameter d sa vyskytuje iba v podmienke (1). Typická hodnota

Obrázok 5: Bohatosť správania Izhikevichovho modelu pri správnom nastavení parametrov



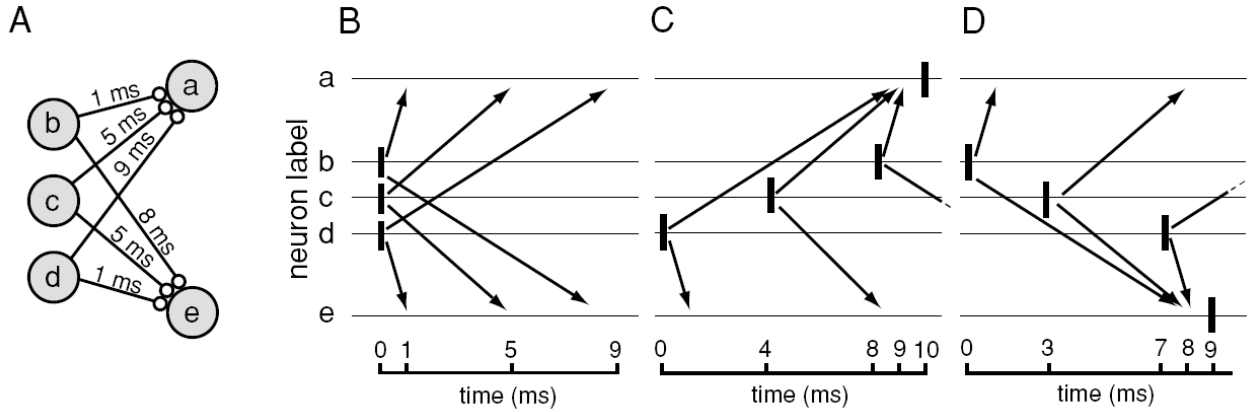
je $d = 2$.

Správnym nastavením týchto parametrov dosiahneme rôzne správanie a sme schopní pokryť dynamiku skoro všetkých typov neurónov v mozgovej kôre. Rôzne správania môžeme vidieť na (obr 5).

2.5.5 Synchronizácia a polychronizácia

Ukazuje sa že neuróny a skupiny neurónov sú schopné generovať časové postupnosti impulzov s viac než milisekundovou presnosťou (pre podrobnosti viď [4]). S týmto zistením sa stále viac pozornosti venovalo možnostiam prenosu informácie pomocou časovania impulzov. Ak dva alebo viac neurónov, ktoré majú spoločný post-synaptický cieľ, vypáli impulz v skoro rovnakom čase bude to mať veľký účinok na post-synaptický neurón. Keby vypálili impulz v rôznych časoch, tak by účinok na post-synaptický neurón bol menší a efekt jednotlivých post-synaptických potenciálov by sa navzájom nemusel sčítať. Tento prístup však predpokladá že prenos impulzu medzi dvoma neurónmi je okamžitý alebo prebieha vo všeobecnosti v skoro rovnakom čase. Čiže predpokladá že informácia sa prenáša synchronizáciou správnych neurónov, ktoré ak sa spolu naraz aktivujú tak vyvolajú potrebnú reakciu v post-synaptickom neuróne.

Obrázok 6: Polychronizácia



(A) Prepojenie neurónov aj s rôznymi oneskoreniami prenosu. (B-D) Postupnosti aktivácie neurónov.

Do výpočtového modelu však môžeme zaradiť rôzne oneskorenia prenosu impulzu axónom a synapsou. Praktické merania ukázali, že rozsah oneskorení v neurónoch mozgovej kôry je veľmi veľký. Oneskorenie medzi dvoma neurónmi môže byť $0.6ms$ až $44.0ms$ (viď [4]). Podľa [4] je rozdelenie jednotlivých hodnôt oneskorení relatívne rovnomerné. Preto sa nám ponúka otázka, prečo sa v mozgu uchováajú takéto rôzne oneskorenia prenosu spolu so schopnosťou prenášať a generovať impulzy so sub-milisekundovou presnosťou, ak pre prenos informácie nie sú dôležité?

Väčšina prác a výskumných smerov zanedbala fakt oneskorenia impulzu pri prenose, považujúc ho iba za bezvýznamnú komplikáciu simulačného výpočtu. Ako vo svojom článku [4] Izhikevich poznamenal, počítanie s oneskoreniami nemusí byť komplikácia. Dokonca tvrdí, že mnohé vzorce správania nie sú bez oneskorení pozorovateľné a model počítajúci s rôznymi oneskoreniami môže mať doteraz nevídané výpočtové a pamäťové možnosti.

Pozrime sa teda na to čo to vlastne je **polychronizácia**. Predstavme si prípad kde neuvažujeme žiadne oneskorenia prenosu impulzu. Majme skupinku niekoľkých neurónov.

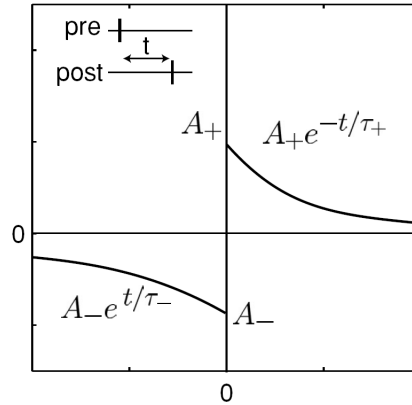
Na to aby boli tieto neuróny schopné aktivovať nejakú ďalšiu skupinu neurónov, museli by páliť naraz aby efekt ich post-synaptických vplyvov bol dostačujúci. Z toho vyplýva, že sú schopné aktivovať (neaktivovať) iba jednu ďalšiu skupinu, tým že budú (nebudú) synchronizované. Ak však berieme do úvahy rôzne oneskorenia pri prenose, tak môžeme dosiahnuť väčšiu pestrosť správania. Je možné túto skupinu naviazať na rôzne skupiny, tak aby pri určitej postupnosti aktivácií raz aktivovali jednu a raz inú skupinu.

Pre bližšie porozumenie si rozoberme prípad na (obr 6). Máme pre-synaptické neuróny **b**, **c** a **d**. A ešte post-synaptické neuróny **a** a **e**. Medzi tým sú synaptické spojenia s rôznymi oneskoreniami (viď. obr 6A). Ak by sa však všetky pre-synaptické neuróny aktivovali naraz, nemuselo by to nič znamenať. Ale ak sa pre-synaptické neuróny aktivujú so správnym načasovaním tak môžu aktivovať neurón **a** (viď 6C). A zároveň s iným načasovaním post-synaptický **b** (viď 6D). Z tohto vidíme, že použitím časových oneskorení sme schopní rozoznať rôzne vzorce načasovania aktivovaním rôznych post-synaptických neurónov (skupín neurónov). Tento fenomén nazveme polychronizáciou, čiže ide o schopnosť prepojenia skupiny s rôznymi post-synaptickými skupinami v závislosti na presnom načasovaní aktivácie jednotlivých neurónov v pre-synaptickej skupine.

Tento mechanizmus Izhikevich použil v náhodnej jednoduchej sieti a prišiel k zaujímavým výsledkom. Zobral náhodnú sieť zloženú z 1000 neurónov. Každé dva neuróny spojil s 10% pravdepodobnosťou. To znamená, že každý neurón bol v priemere spojený so 100 inými neurónmi a každé spojenie malo náhodnú malú váhu. Oneskorenie spojenia nastavil ako náhodnú diskretnú hodnotu od $1ms$ do $20ms$. Pre umiernenie aktivity nastavil 20% neurónov ako inhibičných. Celú simuláciu simuloval v $1ms$ krokoch.

Po 24 hodinách simulovaného času (simulácia trvala 6 hodín), počas ktorých boli váhy modifikované pomocou STDP krivky (obr 7). Počas tejto simulácie boli sledované skupiny neurónov, ktoré boli aktivované zväčša spolu. Za skupinu boli považované neuróny, ktoré boli aktivované vždy keď bola aktivovaná nejaká trojica neurónov a táto trojica vyslala impulzy v určitom presnom načasovaní. Za aktiváciu skupiny sa považoval stav, keď v určitej časovej postupnosti bola aktivovaná väčšina neurónov tejto skupiny. Neuróny vrámci skupiny mali silné spojenia a skupina bola vždy po správnej aktivácii začiatočných

Obrázok 7: STDP krivka



(Spike-Timing Dependent Plasticity). Táto krivka nám hovorí ako pozmeniť váhu synaptického spojenia dvoch neurónov. Keď dôjde k akčnému potenciálu na post-synaptickom neuróne ku ktorému prispel svojim impulzom aj pre-synaptický neurón, tak ich spojenie posilníme. Na to aby k nemu prispel musí impulz z tohto pre-synaptického neurónu doraziť k post-synaptickému neurónu krátky čas pred jeho aktiváciou. Ak však impulz dorazí až po aktivovaní post-synaptického neurónu tak sa jeho vplyv na aktiváciu považuje za zanedbateľný (priam až negatívny) a v tomto prípade dôjde k oslabeniu synaptického spojenia medzi týmito neurónmi.

troch neurónov aktivaná skoro celá. Aktivácia jednej skupiny bola teda považovaná za určitý stav siete. Zaujímavé však je, že počet takýchto skupín v sieti prekročil počet neurónov a pri väčších sieťach dokonca počet synaptických spojení medzi neurónmi. To je možné tak, že každý neurón môže patriť do viacerých skupín a jeho aktivácia sa počíta vždy do inej skupiny v závislosti na kontexte okolitých neurónov.

Zaujímavý fakt je, že každá trojica neurónov mohla definovať viacero skupín v závislosti na načasovaní impulzov. Napriek niektorým otáznym prvkom tohto prístupu ma zaujalo, aké komplexné správanie a komplexný stav siete ponúka jednoduché zavedenie oneskorenia spojenia dvoch neurónov a presného časovania jednotlivých impulzov.

3 Model JASTAP a jeho modifikácie

3.1 Pôvodný model JASTAP

JASTAP je model ktorý je výtvorom pánov J. Janču, I. Stavrovkého a J. Pavláška zo SAV. V porovnaní s inými modelmi impulzných neurónov ide o veľmi jednoduchý model. Napriek tomu je do veľkej miery biologicky akceptovateľný. Je zaraditeľný medzi RSM_0 modely, čiže je založený na pevnej forme funkcie post-synaptického potenciálu. Priebeh tejto presne zadanej funkcie z neho robí zaujímavý model. Ide o priebeh odpozorovaný na živých neurónoch a zároveň vďaka tomuto jednoduchému priebehu sme schopní predvídať správanie modelu, čo sa ukáže pre môj prístup veľmi dôležité. Ako som už spomenul, ide o RSM_0 model, čiže reakcia neurónu na podnet zvonku je vždy rovnaká, nezávisle od hodnoty membránového potenciálu alebo od času posledného vygenerovaného impulzu. Tým sa nám výpočet membránového potenciálu redukuje na sumu post-synaptických potenciálov za určité časové okno, ale to neskôr popíšeme podrobnejšie. Pre viac podrobností o tomto modeli viď [6].

3.1.1 Parametre a vlastnosti modelu

Ďalej si popíšeme parametre modelu tak ako boli pôvodne navrhnuté, neskôr budú popísané zmeny ktoré som navrhol. Vychádzal som z modelu, tak ako bol použitý v prácach [6], [7] a [8].

- **Vstup** Každý JASTAP neurón prijíma vstupy z okolia pomocou synaptických spojení od pre-synaptických neurónov. Je schopný reagovať iba na impulzy, ktoré dorazia na synaptické spojenia. Nie je navrhnutý na prijímanie vstupu vo forme monotónneho prúdu iónov, tak ako tomu je napr. pri modeli Izhikevicha (viď [2]).
- **Post-synaptický potenciál** Funkcia post-synaptického potenciálu má vo všetkých stavoch neurónu rovnaké hodnoty (tvar). Nezávisle na všetkých predchádzajúcich udalostiach vždy na výpočet reakcie membránového potenciálu na prichádzajúci impulz použijeme rovnaký vzťah. Funkcia má tvar:

$$PSP(t) = w \cdot \left(1 - e^{-\frac{t}{t_1}}\right)^2 \cdot e^{-\frac{2t}{t_2}}$$

kde t predstavuje čas, ktorý uplynul od príchodu impulzu na synaptické spojenie post-synaptického neurónu. A parametre t_1 a t_2 určujú časovú dĺžku stúpajúcej a klesajúcej časti krivky funkcie. Tieto parametre by mohli byť aj rôzne pre každú jednu synapsu, ale v tejto práci sú použité ako konštanty pre celý systém. Premenná w je váha synapsy, cez ktorú impulz prišiel.

- **Váha synapsy** Ide o reálnu hodnotu vyjadrujúcu silu (efektivitu) synaptického spojenia, čím je hodnota väčšia v absolútnej hodnote tým je reakcia neurónu na impulz prichádzajúci touto synapsou väčšia. Ak je hodnota kladná ide o excitačné spojenie (excitačný pre-synaptický neurón) a ak je záporná tak ide o inhibičné spojenie (inhibičný pre-synaptický neurón). Hodnota váhy sa môže počas simulácie meniť na základe použitého učenia.
- **Membránový potenciál** Je to hodnota vyjadrujúca veľkosť napätia medzi vnútrom bunky a okolitým prostredím. V modeli JASTAP je to hodnota z intervalu $\langle -1, 1 \rangle$. Získame ju jednoducho ako sumu post-synaptických potenciálov zodpovedajúcim impulzom, ktoré došli za nejaké časové okno. Celková hodnota je potom ešte ohraničená nelineárnou funkciou. Vzťah pre výpočet vyzerá nasledovne:

$$MP(t) = \frac{2}{\pi} \cdot \arctan \left(\sum_{\text{synapsy}} PSP(t) \right)$$

Základná odlišnosť modelu JASTAP v porovnaní s inými modelmi typu SRM_0 je to, že nemá žiadnu funkciu, ktorá by vyjadrovala reakciu na vygenerovanie impulzu. To znamená, že neuvažuje žiadny akčný potenciál. Problém s opätovným generovaním impulzu je ošetrený časovým intervalom, počas ktorého nie je dovolené neurónu znova vygenerovať impulz po tom ako už jeden vytvorí. Tento časový interval nazývame *Absolútna refraktorná fáza*.

- **Aktivačný prah** Je nastaviteľný podľa potreby v konkrétnej aplikácii. Je možné použiť ľubovoľnú hodnotu z $(0, 1)$. Presná hodnota bude vyjadrovať, aké ťažké bude aktivovať daný neurón. Pri hodnote blízkej 0 bude neurón reagovať na hocijaký prichádzajúci kladný impulz. Pri hodnote 1 bude skoro nemožné spustiť vznik impulzu v danom neuróne.

- **Refraktorná fáza** Ide o časový interval, počas ktorého neurón nemože vyslať impulz, aj keď je hodnota membránového potenciálu nad aktivačným prahom. Ale ak sa hodnota MP udrží nad prahom, tak dôjde k vyslaniu impulzu hneď ako skončí daný časový interval. V modeli JASTAP je dĺžka tohto intervalu dynamická, závislá od hodnoty membránového potenciálu:

$$I_a = I_{max} - (I_{max} - I_{min}) \cdot \frac{2}{\pi} \cdot \arctan\left(\frac{MP - \theta}{1 - MP}\right)$$

kde parametre I_{max} a I_{min} vyjadrujú maximálnu a minimálnu dĺžku tohto intervalu. MP je hodnota membránového potenciálu a θ je hodnota aktivačného prahu. Z rovnosti vidieť, že čím je hodnota MP vyššia, tým je interval kratší a tým skôr môže dôjsť k aktivácii (vyslaniu impulzu) znova.

- **Krátkodobé posilnenie synáps** (Short-term facilitation) Pri použití tejto vlastnosti modelu, ktorá bola obsiahnutá v pôvodnom návrhu, sa váha každej synapsy posilní po prechode impulzu touto synapsou. Toto posilnenie je iba krátkodobé a rýchlo vyprchá, potom má už synapsa váhu ako predtým. Ale pri vyššej frekvencii impulzov prechádzajúcich touto synapsou môže dôjsť k dočasnému sčítaniu týchto efektov a značnému posilneniu danej synapsy.
- **Oneskorenie spojenia** Pôvodný model JASTAP neuvažuje žiadne oneskorenie pri prechode impulzu axónom a neskôr synapsou. Prenos impulzov je preto vo všetkých aplikáciách standardného JASTAP-u okamžitý.

3.1.2 Simulácia

Štandardne ako aj u iných modelov sa sieť zložená z neurónov JASTAP simuluje v časových krokoch. Ide o $1ms$ cykly (môže sa použiť aj iná dĺžka krokov), v ktorých sa pre všetky neuróny prepočíta membránový potenciál a ak bol prekročený aktivačný prah vznikne impulz. Tento impulz sa hneď v ďalšom kroku započíta do prislúchajúcich post-synaptických neurónov vo forme post-synaptického potenciálu, ktorý bude ovplyvňovať hladinu MP najbližších pár cyklov.

3.2 Navrhnuté modifikácie

Na začiatok by som rád upozornil na to čo chcelo byť zmenami dosiahnuté, aby čitateľ neskôr pochopil motiváciu, ktorá ma viedla k niektorým rozhodnutiam a konkrétnym zmenám. Cieľom bolo pozmeniť model tak, aby ho bolo možné použiť v simulácii v reálnom čase a nie vo časových krokoch (real-time). To vyžaduje aby sa dalo predvídať správanie jednotlivých neurónov tak, že vieme dopredu zistiť čas v ktorom membránový potenciál prekročí aktivačný prah ak vôbec k prekročeniu má dôjsť. Túto informáciu musíme byť schopní zistiť zo zoznamu udalostí, ku ktorým došlo v neuróne za určité časové okno. Ďalej túto informáciu musíme byť schopní modifikovať na základe toho čo sa ďalej v neuróne udeje.

Ďalej bola snaha zanechať všetky aspekty modelu, ktoré boli biologicky relevantné. Napriek tejto snahe muselo dôjsť k zjednodušeniu v častiach, ktoré by si vyžadovali neunesiteľné výpočtové nároky pri prechode na simuláciu v reálnom čase ak by mali fungovať v pôvodnej forme. Cieľom bolo upraviť model tak, aby čo najviac zodpovedal fungovaniu reálnej biologickej siete.

- **Refraktorná fáza** Ak v neuróne dôjde ku vzniku impulzu tak sa dostane do tohto stavu, počas ktorého nemôže vzniknúť ďalší impulz. Na rozdiel od pôvodného návrhu som refraktornú fázu nepoužil dynamickú. Nie je závislá od hodnoty membránového potenciálu, lebo jeho hodnotu nebudeme počas simulácie sledovať v pravidelných krokoch. Preto by sme nevedeli usúdiť, aká dlhá má byť. Preto tento časový interval použijeme konštantne dlhý, pre všetky neuróny počas celej simulácie.
- **Akčný potenciál** Ako som skôr spomenul, tak pri simulácii model JASTAP priamo neuvažuje žiadnu formu akčného potenciálu. Po prekročení aktivačného prahu na hladinu MP nič okrem samotných PSP nevyplýva. Ja som však automaticky do každého neurónu zabudoval mechanizmus, ktorý nahrádza akčný potenciál tým, že neurón hneď po vygenerovaní impulzu dostane inhibičný impulz, ktorý ho dostatočne hyperpolarizuje. Amplitúda tohto inhibičného impulzu je konštantná, a nastavená podľa potreby pre každú jednotlivú sieť, aby zmiernovala nadbytočnú aktivitu v sieti.
- **Vzťah pre výpočet MP** Zo vzorca pre MP som odstránil časť $2/\pi \cdot \arctan(\dots)$ a zredukoval som ho na $MP(t) = \sum_{\text{synapsy}} PSP(t)$. V simulácii bude hodnota MP

používaná iba na porovnanie s aktivačným prahom preto nemá zmysel ju žiadnym spôsobom dodatočne modifikovať. Stačí modifikovať hodnotu aktivačného prahu, ktorý sa aj tak používa ako konštanta pre všetky neuróny počas celej simulácie.

- **Krátkodobé posilnenie synapsí** Túto vlastnosť modelu som nepoužil vôbec. Napriek tomu že je biologicky opodstatnená, nemá veľa spoločné s prioritným záujmom tejto práce. Vždy sa dá samozrejme do výpočtového modelu pridať, lebo je nezávislá od všetkých jeho ostatných funkcií.
- **Oneskorenie spojenia** Ide o jednu z najdôležitejších vlastností môjho prístupu. Napriek tomu že v pôvodnom modeli nebolo oneskorenie uvažované, v tejto práci je na ňom založených veľa základných vlastností.

Samozrejme hlavnou zmenou je simulácia pomocou udalostí, ktoré sa dejú v reálnom čase s dôrazom na časovú presnosť a nie v časových krokoch. Viac o tom však už v ďalšej kapitole.

3.3 Simulácia v reálnom čase

Skúsme sa zamyslieť nad tým čo z toho čo sa v sieti udeje je naozaj dôležité pre chovanie siete ako celku. Čo vlastne skúmame keď nás zaujíma reakcia siete na nejaký podnet? Vo všeobecnosti jediné praktické údaje, ktoré môžeme zo správania siete vyčítať sú samotné impulzy, ich časovanie a možné zmeny v parametroch modelu ak bol použitý nejaký mechanizmus učenia. Ak ide o väčšiu sieť, tak sa nezaujíname o stav jednotlivých neurónov mimo samotného generovania impulzov v nich a zmien spôsobených učením. Prečo potom sledovať membránový potenciál jednotlivých neurónov, keď je jeho hodnota v rozmedzí, keď daný neurón nemôže ovplyvniť okolie? Táto myšlienka ma priviedla k inej možnosti simulácie neurónov než je obvyklý a dobre fungujúci spôsob simulovania pomocou časových krokov.

Čo tak sledovať iba tie udalosti, ktoré vplývajú na dynamiku samotnej siete. A snažiť sa odhadnúť kedy a či vôbec k týmto udalostiam dôjde. Ak použijeme klasické fungovanie modelu JASTAP, tak jediný spôsob ako môže jeden neurón vplývať na ostatné neuróny v sieti je impulz ktorý vygeneruje. Stačí teda modelovať správanie jednotlivých

neurónov natoľko, aby sme vedeli kedy v nich dôjde k vzniku impulzu. A to je možné aj bez toho aby sme sledovali hladinu membránového potenciálu v pravidelných časových krokoch.

Zavedieme si teda udalosť. Každá udalosť bude niesť informáciu kedy nastala (má nastať), čo presne sa stane a ktorého neurónu sa udalosť týka. Idea funkcie samotnej udalosti je mienená tak, že počas výpočtu v správny čas nejakému neurónu oznámi že sa s ním dačo stalo. Udalosti môžeme podľa informácie, ktorú nesú nasledovne rozdeliť:

- **Dorazil impulz** Udalosť hovorí o tom, že do daného neurónu dorazil impulz z niektorého z jeho pre-synaptických susedov. Neurón následne započíta post-synaptický potenciál vyvolaný týmto impulzom do výpočtu svojho stavu.
- **Bol prekročený prah** Táto udalosť neurónu hovorí, že ak sa od času vzniku tejto udalosti nič nestalo, tak by mal membránový potenciál prekročiť aktivačný prah. Za vznik udalosti považujeme čas simulácie, kedy sme sa dopočítali že ak sa ďalej nič neočakávané nestane, tak v neuróne dôjde k prekročeniu prahu.
- **Neznámy stav** Táto udalosť hovorí, že dynamika daného neurónu nebola ďalej skúmaná a treba zistiť, či sa neurón neblíži k prekročeniu prahu. Tento druh udalosti má zmysel, keď počas výpočtu presného času prípadného prekročenia prahu zistíme, že k prekročeniu nedôjde v najbližšom čase. Vtedy už nemusí mať zmysel ďalej mýňať čas počítaním, ak je veľká šanca že sa objaví udalosť, ktorá aj tak zmení presné správanie neurónu.
- **Koniec refraktornej fázy** Skončila refraktorná fáza a treba zistiť, či je MP nad AP a má vzniknúť nový impulz alebo nie.

Kedy a ako presne jednotlivé udalosti vznikajú a čo sa vykoná keď na ne príde čas je popísané nižšie. Skúsme teraz navrhnúť mechanizmus fungovania takejto simulácie v reálnom čase. Rozdeľme ho na niekoľko nezávislých častí.

- **Manažovanie udalostí** Čo všetko brať ako udalosť? Ako spravovať jednotlivé udalosti? Kde ich uložiť keď ich získame? Čo s udalosťami, ktoré už nie sú aktuálne?
- **Výpočet membránového potenciálu** Ako vypočítať hodnotu membránového potenciálu v presnom časovom okamihu? Dá sa hodnota rýchlejšie a hrubo odhadnúť, keď nám stačí vedieť či je nad alebo pod určitou hranicou?

- **Výpočet času vzniku impulzu** Kedy dôjde v neuróne ku vzniku impulzu? Môže s doterajšími vstupmi k prekročeniu aktivačného prahu vôbec dôjsť?

3.3.1 Manažovanie udalostí

V tejto prvej časti je obsiahnutý mechanizmus uskladnenia, zoraďovania, hľadania a realizácie jednotlivých udalostí. Základný dôraz sa berie na rýchlosť zaradenia novej udalosti do radu udalostí zoradeného podľa času, kedy sa majú vykonať. V tomto prípade už nebude problém nájsť udalosť, ktorá sa má vykonať ako prvá a po jej vykonaní vyradiť ju zo zoznamu. Alternatívou je viac času venovať hľadaniu najbližšej udalosti na vykonanie a zoraďovaním sa veľmi netrápiť. Toto by sa dalo implementovať pomocou nejakej formy hašovania. Z týchto dvoch možností som vybral určitý mix.

Použil som tzv. **prioritný rad** (priority queue), je to prístup v ktorom pri pridaní každej ďalšej položky do zoznamu pridávame prvok do zoradenej postupnosti tak, aby sme zoradenie neporušili. Ide však iba o abstraktný prístup, ktorý musí byť konkrétnym spôsobom implementovaný. Pri rôznych možnostiach bolo treba hlavne hľadiť na časovú zložitosť danej implementácie.

Prvou možnosťou implementácie bol **spájaný zoznam**. V tomto prípade je však obtiažne nájsť správne miesto na vloženie práve vzniknutej udalosti. Pri veľkom počte už uskladnených udalostí by hľadanie miesta trvalo príliš dlho.

Druhou možnosťou bolo **dynamické pole** v ktorom, ak v jednom políčku poľa iba jeden prvok, je veľký problém samotné vloženie prvku medzi už existujúce udalosti. Taktiež nájsť miesto kam novú udalosť vložiť je tiež časovo náročné.

Rozhodol som sa preto pre trochu komplexnejšie riešenie. Založil som štruktúru na poli. Konkrétne na statickom poli, ktoré predstavovalo určité časové okno od aktuálneho simulovaného času do budúcnosti. Toto okno sa samozrejme posúvalo s plynúcim simulovaným časom. Okno muselo byť dostatočne dlhé na to aby sa minimalizovala šanca, že vznikne udalosť, ktorá sa má vykonať až za koncom tohto okna. Každé políčko tohto poľa predstavovalo nejaký pevný časový interval a obsahovalo jedno dynamické pole, ktoré ob-

sahovalo samotné udalosti patriace vykonaním do tohto časového intervalu. Dĺžka týchto časových intervalov bola samozrejme nastaviteľná a počas pokusov sa pri optimalizácii ustálila na 10^{-5} sekundy. Samozrejme pri väčších sieťach, kde sa predpokladá väčší počet udalostí na časovú jednotku by bolo vhodné použiť kratšie intervaly. Menením dĺžky intervalu sme schopní presunúť časť z výpočtovej zložitosti do pamäťovej a naopak. Pri vzniku novej udalosti sa v konštantnom čase dá nájsť časový interval, v ktorom by mala daná udalosť byť. Tá je následne pridaná do dynamického poľa udalostí pre daný interval tak, aby bolo toto dynamické pole stále usporiadané. Udržiavať usporiadanie tohto poľa nie je obtiažne, lebo je snaha nastaviť veľkosť intervalov tak, aby na jeden interval pripadal minimálny počet udalostí. Ak sa predpokladá že intervaly budú dostatočne malé a pravdepodobnostné rozdelenie udalostí bude v čase náhodné, tak by pridanie mohlo skončiť v skoro konštantnom čase. Taktiež ako z konštrukcie vidieť, tak nájdenie prvej udalosti na vykonanie je zvládnuteľné v konštantnom čase. Nie je ani potrebné už vykonanú udalosť vyhadzovať z pamäte, stačí si pamätať nejaký ukazovateľ, ktorý bude ukazovať kde sme skončili a pokračovať sa bude na ďalšej udalosti. Keď sa potom vyčerpajú udalosti z celého jedného časového intervalu, tak ho môžeme vyprázdniť celý naraz. A to tiež bez časových nárokov.

Jediný problém môže nastať ak vznikne udalosť, ktorá sa má vykonať mimo časového okna, ktoré máme zatiaľ pripravené. Túto možnosť som riešil tak, že som použil náhradnú udalosť, ktorú som uložil náhodne niekam na koniec časového okna ktoré bolo zatiaľ rezervované. Táto udalosť niesla informáciu, že niečo sa malo vykonať až neskôr. A jej vykonanie bolo nakoniec v tom, že obnovila pôvodnú udalosť, ak sa už časové okno posunulo dostatočne dopredu. Inak vytvorila ďalšiu náhradnú udalosť znova na konci časového okna, zase o kus neskôr, postupne sa približujúc k času vykonania pôvodnej udalosti.

Veľa druhov udalostí nepotrebovalo túto náhradnú udalosť. Napríklad, keď došlo k vzniku impulzu v neuróne a impulz sa začal šíriť axónom ďalej do synaptického spojenia. Tak vznikli udalosti hovoriace o dorazení impulzu do všetkých post-synaptických neurónov s ktorými bol daný neurón spojený. A o tých udalostiach vieme povedať na isto to, že môžu vzniknúť pred svojím vykonaním iba toľko, aké je najdlhšie oneskorenie v prenose impulzu

v celej sieti. Dĺžky týchto oneskorení sú ohraničené a stačí časové okno v prioritnom rade nastaviť tak, aby sa udalosť tohto typu vždy vošla do časového okna.

3.3.2 Výpočet membránového potenciálu

Najdôležitejšou časťou celého algoritmu je výpočet hodnoty membránového potenciálu. Táto časť zaberá najviac výpočtového času a preto najväčšia snaha o optimalizáciu bola venovaná práve tu. Práve tu sa však z môjho pohľadu prejavila najväčšia nevýhoda modelu JASTAP vzhľadom na moje zámery. A to je spôsob akým sa počíta hodnota membránového potenciálu. Pozrime sa však na tento výpočet zblízka.

Ako bolo už skôr spomenuté, v modeli JASTAP je hodnota membránového potenciálu vyjadrená ako suma všetkých post-synaptických potenciálov, ktoré ešte stále ovplyvňujú jeho hodnotu. Pre zopakovanie si ju napíšme ešte raz.

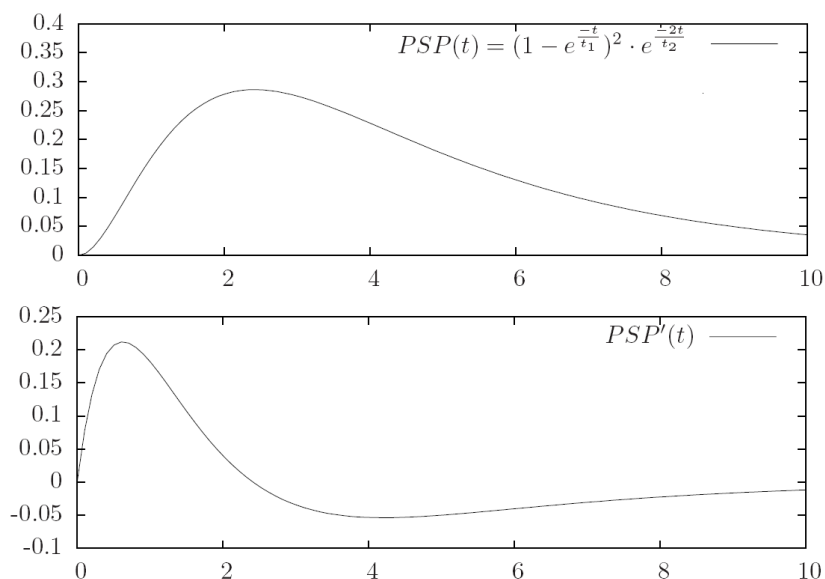
$$MP(t) = \frac{2}{\pi} \cdot \arctan \left(\sum_{\text{synapsy}} PSP(t) \right)$$

Najjednoduchší spôsob je teda zosumovať všetky PSP. To by ale mohlo byť problematické. Podľa údajov uvedených v [1] je pre priemerný neurón treba rádovo 50 dobre načasovaných impulzov na to, aby spôsobili prekročenie aktivačného prahu. Z tejto informácie môžeme usúdiť že ak budeme simulovať biologicky relevantnú sieť, tak bude treba sumovať rádovo 100 nezávislých PSP. Počas mojich pokusov sa však počet PSP, ktoré v jednom okamihu ovplyvňovali jeden neurón nakopilo oveľa viac, až okolo 1000. A aj keď je snaha použiť tento model na simuláciu sietí s biologicky relevantnými parametrami, kde tento počet až tak nestúpne, je sumovanie mnohých PSP vážny problém ktorý treba riešiť.

Napriek mnohým snahám nahradiť sumovanie všetkých post-synaptických potenciálov niečím iným som na lepší spôsob neprišiel. Keď sa pozrieme na tvar funkcie PSP a jej derivácie (obr 8), tak vidíme že ide o jednoduché a ľahko predvídateľné funkcie. Jeden z pokusov ako vyriešiť tento problém bola snaha aproximovať funkciu PSP polynómom.

Prvý nápad ako zlepšiť túto situáciu bol, aproximovať funkciu PSP polynómom 5-7 stupňa (stupeň by sa nastavil podľa toho či chceme dôraz na presnosť alebo efektívnosť

Obrázok 8: Funkcia PSP a jej derivácia



algoritmu) aspoň na intervale na ktorom nás hodnota tejto funkcie zaujíma. Ďalej by nebolo problematické priamo do polynómu zakomponovať váhu synaptického spojenia. Predstava bola taká, že by sa každý PSP (už vynásobený váhou svojej synapsy) vystupujúci v sume nahradil polynómom. Dostali by sme nasledovný polynóm:

$$PSP(t) = a_7t^7 + a_6t^6 + a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$$

Tieto polynómy by sa potom zosumovali iba raz do jedného celkového polynómu reprezentujúceho hodnotu MP a pri výpočte hodnoty celkového MP by stačilo vyčísliť tento jeden celkový polynóm. Mechanizmus mohol fungovať tak, že keď dorazil nový impulz a vyvolal PSP, tak sme príspevok tohto PSP aproximovali polynómom a následne pripočítali do celkového polynómu. Pri pripočítavaní polynómov však bolo treba samozrejme myslieť na to, že sú navzájom časovo posunuté a preto ich treba prepočítať, tým že premennú t v nich nahradíme výrazom $t - s$, kde s je hodnota časového posunu nového polynómu oproti polynómu vyjadrujúcejmu MP. Pri tomto prístupe by nás to iba stálo pre každý PSP iba raz daný polynóm pripočítať a po uplynutí času, v ktorom nás príspevok tohto PSP zaujímal polynóm odpočítať. A zistenie hodnoty v danom čase by nás stálo konštantne veľa času.

Tento zo začiatku veľmi sľubný prístup však narazil na vážny a bohužiaľ neprekonateľný problém. Tichým predpokladom úspechu polynómového prístupu je fakt, že časy v ktorých nás bude zaujímať hodnota celkového membránového potenciálu budú postupne stúpať. Čo však vôbec nebolo v praxi pravda. Prakticky vždy počas simulácie je potrebné skákať v čase (v ktorom sa zaujímate o hodnotu MP) dopredu a dozadu. A to pri relatívne drahých výpočtoch potrebných pre odčítania a pričítanie polynómov prejaví na veľkej výpočtovej zložitosti. Na potrebu zisťovať hodnotu MP v rôznych, nie po sebe idúcich časoch sa ešte bližšie pozrieme v ďalšom odseku.

Napriek tomu že som tento prístup nakoniec nepoužil je zaujímavý. Je možné že pri určitej úprave celého systému by mohol byť veľmi praktický. Vďaka nemu sa môžeme zbaviť potreby stáleho sumovania funkcií za predpokladu, že vieme zabezpečiť jednosmerné plynutie času pri zisťovaní hodnôt MP.

Druhý prístup ktorý som použil bol trochu všeobecnejší. Keď som sa nevedel vyhnúť sumovaniu všetkých PSP pri zisťovaní hodnoty MP v konkrétnom čase, tak som sa rozhodol toto sumovanie zrýchliť. Keď sa nemôžeme vyhnúť prejdenu cez všetky PSP, tak sa môžeme aspoň pokúsiť zrýchliť výpočet každého z nich. Preto som sa zameral na čo najrýchlejší spôsob ako vypočítať hodnotu PSP v určitom čase. Funkcia post-synaptického potenciálu ma relatívne komplikovaný tvar:

$$PSP(t) = w \cdot \left(1 - e^{-\frac{t}{\tau_1}}\right)^2 \cdot e^{-\frac{2t}{\tau_2}}$$

Nie je až také výpočtovo náročné vyčíslíť hodnotu, ale napriek tomu sa tu dá nejaký čas ešte získať. Keď si na tomto mieste uvedomíme, že celkový algoritmus pracuje iba s určitou vpred nastavenou presnosťou. Povedzme napríklad, že bola nastavená presnosť 10^{-5} sekundy (ide o akceptovateľnú odchýlku pri zisťovaní času vzniku impulzu). Pri tejto nastavenej presnosti nepotrebujeme hodnotu PSP až tak presne, stačí preto keď si ju prepočítame dopredu. Povedzme s rozlíšením 10^{-5} sekundy. Za predpokladu, že dĺžka pre nás zaujímavej časti vlny PSP je asi 10 ms. Čo v našom prípade činí 1000 predpočítaných hodnôt. Samozrejme nie je problém rozlíšenie s ktorým si hodnoty predpočítame rapídne zväčšiť. Pri terajších možnostiach je jednoduché držať v pamäti

aj 10^6 hodnôt. Týmto spôsobom je možné zistiť hodnotu PSP v ľubovoľnom bode relatívne rýchlo a s dostatočujúcou presnosťou. A stačí nám na to jednoduchý výpočet indexu predpočítaného poľa hodnôt z hodnoty času a samotný prístup do poľa na daný index.

Rovnaký spôsob som aplikoval aj na deriváciu funkcie post-synaptického potenciálu. Ide o podobnú funkciu, ktorej vyjadrenie je iba o trošku zložitejšie. Taktiež rovnakým spôsobom pristupujem k hodnotám derivácie jednotlivých PSP ako aj k derivácii celej sumy, čiže aktuálnej derivácii hodnoty membránového potenciálu. Na čo sú hodnoty derivácií dobré viac vysvetlím v ďalšom odseku. Budú hlavne použité na urýchlenie numerických metód použitých na odhad času v ktorom membránový potenciál prekročí aktivačný prah.

3.3.3 Výpočet času vzniku impulzu

Posledná a najdôležitejšia časť algoritmu je aproximačný výpočet času prekročenia aktivačného prahu. Ide o časť kde sa dá správnym prístupom dosiahnuť najväčšie urýchlenie výpočtu. V tejto časti dochádza ku vzniku samotných udalostí a zároveň aj k ich vykonaniu. Spíšme si teda čo je úloha tejto časti algoritmu a popíšme jeho fungovanie. Ku každému druhu udalosti je popísané čo sa vykoná a aké prípadné nové udalosti vyplynú z výsledkov.

Udalosť: Dorazil impulz do neurónu

- Pridáme informácie o post-synaptickom potenciáli spôsobenom týmto impulzom medzi ostatné post-synaptické potenciály ktoré si daný neurón pamätá. Čiže každý neurón má zoznam všetkých PSP, ktoré vznikli za nejaký čas. Pamätá si iba tie ktoré reálne vplývajú na hodnotu jeho MP. Tie ktoré už odznievajú a ich vplyv je už minimálny vyhodíme zo zoznamu. Ku každému si zapamätáme čas jeho začiatku, teda čas kedy dorazil impulz ktorý ho spôsobil. Ďalej si ešte musíme zapamätať váhu synapsy ktorou prišiel v momente keď dorazil. Túto váhu potrebujeme, keď budeme vyčíslávať jeho vplyv na PSP.
- Ako ďalší krok si pozrieme, či sa neurón náhodou nenachádza v refraktornej fáze, v ktorej určite nemôže dôjsť ku vzniku impulzu po prekročení aktivačného prahu. Ak neurón je v refraktornej fáze, tak nemá zmysel nič ďalej simulovať a môžeme sa

presunúť na ďalšiu udalosť. Ak však nie je, tak pokračujeme ďalej v tejto postupnosti krokov.

- Zistíme, či je možné aby post-synaptický potenciál vyvolaný týmto impulzom spôsobil prekročenie aktivačného prahu. Ide o veľmi jednoduchý a rýchly test, ktorý sa vykoná vždy po príchode impulzu do neurónu. Udržiavame si hodnotu vyjadrujúcu sumu všetkých amplitúd PSP v zozname. Za každý PSP pripočítame jeho maximálnu hodnotu keď začne pôsobiť. Výpočet maximálnej hodnoty je triviálny. Keďže tvar a hodnoty každého PSP sú ekvivalentné, keď neberieme ohľad na váhu synapsy. Stačí teda iba na začiatku raz zistiť maximum pôvodnej funkcie PSP a potom pre každý ďalší impulz vynásobíme túto hodnotu váhou synapsy a pripočítame do našej sumy. Po skončení vplyvu PSP na MP, vyhodíme daný PSP zo zoznamu a s ním odpočítame jeho maximum zo sumy. Samotný test spočíva potom v porovnaní aktivačného prahu a sumy maxím. Ak je suma menšia tak nemá zmysel ďalej nič dopočítavať, lebo nie je možné aby došlo k prekročeniu aktivačného prahu. V tomto prípade môžeme prejsť na ďalšiu udalosť. Ak je však suma väčšia, tak sa ďalej snažíme zistiť čo sa v tomto neuróne stane. Tieto úvahy platia, ak ide o excitačný PSP. V prípade že ide o ihibičný PSP tak považujeme za max hodnotu nula.
- Ak ani predošlý test nezistí, že nie je možné dosiahnuť aktivačný prah musíme použiť náročnejšie prostriedky. Dobrá správa však bola, že iba málo impulzov prešlo týmto prvým testom. To znamená že veľa impulzov nie je dôležitých a ich vplyv sa skúma až neskôr, keď dorazia ďalšie impulzy, ktoré už v spojení s predošlými môžu aktivovať daný neurón. Ak teda predošlý test povie, že môže dôjsť k prekročeniu prahu ďalším krokom bude iteračný aproximačný algoritmus, ktorý zistí konkrétny čas prípadného dosiahnutia prahu. Tento algoritmus je opísaný v ďalšej podkapitole.
- Sú tri možné výsledky, ktoré môžeme dostať z aproximačného algoritmu. Zistí, že **nedôjde k prekročeniu prahu**. Druhá možnosť je, že **zistí čas prekročenia** s danou presnosťou. Alebo usúdi, že **nedôjde k prekročeniu v najbližšom čase**, ale túto možno nevytlúči pre budúcnosť.
- Ak nedôjde k prekročeniu prahu, tak vieme že práve skúmaná udalosť už nič ne-
spôsobí. Presunieme sa teda na ďalšiu udalosť v našom prioritnom rade.

- Ak zistí presný čas prekročenia prahu, tak vytvoríme novú udalosť "Vznik impulzu" hovoriacu o tom, že v danom neuróne dôjde v tom presnom čase k **vzniku impulzu**. Následne môžeme tiež prejsť na ďalšiu udalosť ktorú máme v prioritnom rade.
- Ak nemá dôjsť k prekročeniu aktivačného prahu v určitom časovom intervale, ale nevieme či dôjde k prekročeniu neskôr tak musíme vytvoriť nejaké upozornenie. Upozornenie, ktore nám neskôr povie, že treba preskúmať správanie tohto neurónu neskôr, či náhodou MP nedosiahne aktivačný prah. Na tento účel si vytvoríme ďalší druh udalosti, nazveme ho "Treba prepočítať správanie MP".

Udalosť: Vznik impulzu

- Prvé čo musíme zistiť, keď príde udalosť "Vznik impulzu" je, či sa niečo v danom neuróne od jej vzniku zmenilo. Ak by napr. do neurónu od vzniku tejto udalosti dorazil ďalší impulz, či už excitačný alebo inhibičný tak už táto udalosť nie je aktuálna. Pretože čas v ktorom prekročí neurón aktivačný prah sa určite posunul, ak ešte vôbec dôjde k prekročeniu prahu. Túto situáciu môžeme vo všeobecnosti riešiť tak, že po vzniku udalosti "Vznik impulzu" si nastavíme do pomocnej premennej identifikačné číslo (alebo niečo podobné na identifikáciu) danej práve vzniknutej udalosti. Keď potom do neurónu dorazí ďalší impulz, teda nastane udalosť "Dorazil impulz do neurónu" pre tento neurón tak musíme nejakým spôsobom zaznačiť, že došlo k zmene plánov. Ak sa po novom impulze posunie čas prekročenia prahu, tak vytvoríme novú udalosť "Vznik impulzu" i prepíšeme zapamätané identifikačné číslo. Ak po zmene k prekročeniu prahu vôbec nedôjde, tak jednoducho identifikačné číslo vynulujeme. Takto budeme ľahko vedieť, či udalosť "Vznik impulzu" ktorá práve došla je aktuálna, alebo bola už prepísaná inou podobnou udalosťou.
- Ak udalosť nie je aktuálna tak na ňu môžeme zabudnúť a ísť na ďalšiu udalosť.
- Ak je udalosť stále aktuálna tak naozaj došlo k prekročeniu aktivačného prahu a môžeme vygenerovať samotný impulz. To znamená, že vytvoríme pre každý postsynaptický neurón spojený s týmto neurónom udalosť "Dorazil impulz do neurónu". Každá udalosť musí viesť cez ktorú synapsu impulz ktorý reprezentuje prešiel, aby bolo možné aplikovať silu váhu danej synapsy. Najdôležitejšia vec je však zobrať

do úvahy oneskorenie synaptických spojení. Pre každú novovzniknutú udalosť podľa oneskorenia danej synapsy vypočítame čas kedy sa má vykonať, tým že k času vzniku impulzu jednoducho pripočítame oneskorenia jednotlivých spojení.

Udalosť: Treba prepočítať správanie MP

- Tu ako aj v predošlom prípade treba najskôr použiť mechanizmus na rozlíšenie stále aktuálnych udalostí od tých ktoré už nie sú dôležité. Je možné použiť tú istú premennú na pamätanie identifikácie poslednej aktuálnej udalosti. A to preto lebo je vždy iba jedna aktuálna udalosť, vždy keď vznikne nová, či už ide o "Vznik impulzu" alebo "Treba prepočítať správanie MP" tak všetky predošlé nás už nemusia zaujímať.
- Ak udalosť nie je aktuálna tak na ňu môžeme zabudnúť a ísť na ďalšiu udalosť.
- Ak je udalosť stále aktuálna tak spustíme výpočet času, kedy dôjde k prekročeniu aktivačného prahu. A rovnako ako v prípade udalosti "Dorazil impulz do neurónu" začneme elementárnym testom a neskôr ďalej úplne rovnako ako v tom prípade.

Udalosť: Koniec refraktornej fázy

- Udalosť hovorí o tom, že skončil čas počas ktorého neurón nemohol vyslať impulz. Ako prvé nastavíme premennú ktorá určuje či je neurón v refraktornej fáze na *false*, aby v ďalších výpočtoch vedel že už nie je. Jediné čo je treba urobiť ďalej, je zistiť akú hodnotu má MP teraz.
- Ak je MP nad aktivačným prahom tak vytvoríme impulz presne rovnako ako na konci v prípade udalosti typu "Vznik impulzu". Nič iné nie je potrebné simulovať alebo zisťovať.
- Ak MP nie je nad aktivačným prahom tak spustíme výpočet, ktorý zistí ako sa bude MP ďalej vyvíjať a vykonáme príslušné kroky. Presne rovnako ako v prípade udalosti typu "Dorazil impulz do neurónu"

3.3.4 Aproximačný iteračný algoritmus

V tejto časti sa pokúsím popísať algoritmus, ktorý sa postupných iteráciách snaží zistiť čo najpresnejšie kedy presne a či vôbec v neuróne membránový potenciál prekročí aktivačný prah a dôjde k vzniku impulzu. Ako som už v skôr v kapitole spomenul, zistenie presnej hodnoty MP v konkrétnom simulovanom čase je drahá operácia. Preto bude snaha obmedziť potrebu vedieť presnú hodnotu. V algoritme je použitých aj veľa ďalších prístupov v snahe urýchliť výpočet. Mnohé z nich sú založené na predpočítaní potrebných hodnôt a ich zápise do poľa / tabulky aby sme mohli ich hodnoty priamo čítať a nemuseli opätovne počítať. Začnime teda zoznamom predpočítaných hodnôt a parametrov použitých v algoritme.

Parametre a predpočítané konstanty

- δ - Rozlíšenie na predpočítanie hodnôt PSP a iných funkcií.
- $PSP(t)$ - Predpočítané pole hodnôt post-synaptického potenciálu, počet prvkov poľa by mal byť $\delta \cdot \Delta t$, kde Δt je dĺžka časového intervalu na ktorom má PSP zaujímavé hodnoty (dostatočne rôzne od nuly) v sekundách.
- $PSP'(t)$ - Predpočítané pole hodnôt derivácie funkcie PSP, podobné ako $PSP(t)$.
- t_d - Je časová konštanta / parameter, ktorá vyjadruje určitú časť priebehu vývoja PSP. Jej význam bude podrobne vysvetlený neskôr, ale je dôležité si všimnúť jej význam pri tvorbe ďalších predpočítaných dát.
- $PSP_d(t)$ - Matematicky zapísané: $PSP_d(t) = \max\{PSP'(t^*) \mid t^* \in \langle t, t + t_d \rangle\}$. Ako v matematickom zápise vidieť, hodnota PSP_d v určitom čase hovorí aká je najväčšia derivácie PSP v intervale napravo od tohto času s dĺžkou t_d . Význam týchto údajov bude tiež vysvetlený neskôr.
- α - Konštanta. Vyjadruje čas v ktorom PSP dosiahne svoje maximum. Matematicky zapísané: $PSP(\alpha) = \max\{PSP(t) \mid t \in \langle 0, \infty \rangle\}$.

Priebeh algoritmu

Pozrime sa teraz ako prebieha samotný algoritmus. Na začiatku inicializujeme hodnotu

t_0 na čas, v ktorom sa mala vykonať udalosť spúšťajúca tento výpočet. V tomto čase vieme, že hodnota membránového potenciálu je pod aktivačným prahom. Ak by nebol, tak by k spusteniu algoritmu vôbec nedošlo, lebo by sa výpočet zastavil už na teste ktorý predchádza tento algoritmus a prešlo by sa na ďalšiu udalosť. Počas behu algoritmu bude hodnota t_i postupne iterovať k času prekročenia aktivačného prahu alebo rýchlo preletí za časový interval ktorý nás zaujíma.

Na začiatku si však treba zvoliť čas, ako dlho budeme simulovať správanie MP než vyhlásime, že ďalej nás už nezaujíma. Voľba tohto času je otázkou stratégie, ktorú chceme použiť, lebo táto zvolená hodnota značne vplýva na vlastnosti algoritmu. Algoritmus môže skončiť nájdením prekročenia aktivačného prahu ale taktiež nemusí. Je už na nás kedy ho zastavíme, keď sa mu nedarí nájsť dosiahnutie aktivačného prahu. Jedna možnosť je odsimulovať vývoj MP až po koniec posledného PSP. V tomto prípade si budeme istí, že do teraz prišlé impulzy určite nespôsobia depolarizáciu potrebnú na vznik impulzu ak algoritmus nenašiel prekročenie prahu. Ale je tu veľké nebezpečenstvo zbytočného počítania. Ak simulujeme veľký časový úsek, tak sa zväčšuje pravdepodobnosť, že v danom časovom úseku dorazí do neurónu ďalší impulz a nami simulovaný vývoj už nebude aktuálny. Preto je dobré zvoliť si určitú časovú dĺžku, ktorú budeme simulovať a ak počas nej nedôjde k prekročeniu, tak si časový okamih v ktorom sme skončili označíme novou udalosťou typu "Treba prepočítať správanie MP" a môžeme prejsť na ďalšiu udalosť. Týmto minimalizujeme pravdepodobnosť, že niečo budeme počítať zbytočne. Keď potom dorazí udalosť "Treba prepočítať správanie MP" v neuróne sa nič nové neudialo, tak sa jednoducho pokračuje v algoritme. Ak sa však zmenilo, tak už nemá zmysel skúmať správanie membránového potenciálu, lebo iniciatívu už prebrala iná udalosť, ktorá bude za toto zodpovedná.

Samotné výpočtové iterácie sú založené na nasledovnom pravidle. Pre presný popis premenných a parametrov v rovnici pozrite prehľad uvedený vyššie.

$$t_{i+1} = t_i + \max\left\{\frac{1}{\delta}, \min\left\{t_d, \frac{\theta - \text{MP}(t_i)}{\text{PSP}_d(t_i)}\right\}\right\}$$

Pre bližšie pochopenie vzťahu vysvetlím ako vznikol a prečo je práve takýto. Ako som

už spomenul, základným predpokladom začiatku algoritmu je, že t_0 je čas vykonávania udalosti, ktorá spustila algoritmus. A druhým rovnako dôležitým predpokladom je, že hodnota $MP(t_0)$ je menšia ako hodnota aktivačného prahu θ .

Algoritmus iteruje dovtedy, kým hodnota $MP(t_i)$ nepresiahne aktivačný prah, alebo kým čas t_i nepresiahne hranicu, po ktorú sme sa rozhodli simulovať správanie neurónu.

Popíšme si teraz jeden základný iteračný krok algoritmu. Začíname v čase t_i , o ktorom vieme že v ňom hodnota MP nepresahuje aktivačný prah θ . Na tomto mieste využijeme informáciu o maximách derivácií pre intervaly začínajúce v určitom čase, čiže použijeme pole $PSP_d(t)$. To nám hovorí, aká je najväčšia derivácia jedného PSP na intervale od času t s dĺžkou t_d . Keď zosumujeme tieto hodnoty pre všetky PSP, ktoré máme pre daný neurón zapamätané tak dostaneme maximum možnej derivácie celého MP pre daný interval. A keď vieme, aká je hodnota MP v časovom bode t a ďalej vieme aká je maximálna derivácia na časovom intervale $\langle t, t + t_d \rangle$ potom vieme vypočítať maximálnu hodnotu akú môže MP dosiahnuť na tomto intervale.

- Ak je táto maximálna hodnota menšia ako θ . Tak potom na tomto intervale určite nedôjde k prekročeniu aktivačného prahu. Prejdeme preto k ďalšej iterácii, čiže k iterovanému času pripočítame dĺžku intervalu ($t_{i+1} = t_i + t_d$). Ak sa t_{i+1} stále nachádza v časovej oblasti o ktorú sa zaujíname, tak pokračujeme ak nie tak skončíme algoritmus s tým, že sme získali informáciu, že do času t_{i+1} nedošlo k prekročeniu prahu.
- Ak je táto hodnota väčšia ako θ . Tak na skúmanom intervale môže dôjsť k prekročeniu prahu. Ale vďaka informácii o derivácii a hodnote MP vieme určiť kedy najskôr môže dôjsť k prekročeniu prahu. Vieme hodnotu derivácie $PSP_d(t)$ a to, že membránovému potenciálu chýba do prahu ešte $\theta - MP(t_i)$. Z toho môžeme usúdiť, že ak dôjde k prekročeniu aktivačného prahu tak to bude určite neskôr ako v čase $t_i + (\theta - MP(t_i)) / (PSP_d(t_i))$. K prekročeniu určite nedôjde presne v tomto bode, lebo funkcia PSP nemá na žiadnom netriviálnom intervale konštantnú deriváciu. Tu však ešte musíme rozlíšiť dva prípady.

– Ak $(\theta - MP(t_i)) / (PSP_d(t_i)) \geq \frac{1}{8}$ tak stačí, keď nastavíme nový čas na hodnotu

o ktorej vieme, že hodnota MP je tu stále ešte menšia ako aktivačný prah. V tomto prípade teda: $t_{i+1} := t_i + (\theta - MP(t_i)) / (PSP_d(t_i))$.

- Ak však platí $(\theta - MP(t_i)) / (PSP_d(t_i)) < \frac{1}{\delta}$ tak by mohlo hroziť, že síce čas by sa posunul, ale z predpočítaných polí by sme načítali rovnaké hodnoty ako v čase o krok späť. Preto musíme zariadiť aby sa čas posunul aspoň o rozlišovací diel predpočítaných polí, čo je $\frac{1}{\delta}$. To je ale jediná vetva algoritmu kde **môže hodnota $MP(t_{i+1})$ prekročiť alebo aspoň dosiahnuť aktivačný prah**. Tým že sme čas posunuli viac ako nám dovoľovala podmienka zaručujúca, že $MP(t_{i+1})$ bude pod aktivačným prahom.

Iterácie mohli dopadnúť dvomi spôsobmi. Buď sme našli čas prekročenia aktivačného prahu, alebo sme prešli celý časový interval, ktorý nás zaujímal. Pre oba možné konce sú udalosti, ktoré treba vytvoriť:

- Vyšli sme simulačným časom z intervalu, ktorý nás zaujímal a stále sme neprekročili prah. Vtedy si daný čas označíme udalosťou "Treba prepočítať správanie MP" a môžeme pokročiť na vybavenie ďalšej udalosti.
- V čase t_{i+1} prekročili aktivačný prah. V tomto prípade vieme, že hodnota $MP(t_i)$ je stále menšia ako aktivačný prah. To vieme usúdiť z toho, že ak dôjde k prekročeniu prahu tak sa iterovanie ukončí a iba pre posledný časový bod platí $MP(t_{i+1}) > \theta$. Zároveň vieme, že časy t_i a t_{i+1} predstavujú susedné políčka predpočítaných polí. To vieme preto, lebo k prekročeniu prahu môže dôjsť iba v poslednej vetve algoritmu, kde sa čas posúva presne o $\frac{1}{\delta}$. Máme preto dve vedľajšie hodnoty MP, jednu nad a druhú pod aktivačným prahom. Tu už lineárne odhadneme, kde medzi týmito dvoma bodmi došlo k prekročeniu prahu. Tento lineárny odhad nie je potrebný aby sme splnili presnosť ktorú sme si predpísali, ale získame takto oveľa presnejšiu hodnotu. Následne vytvoríme udalosť typu "Vznik impulzu" pre daný čas a neurón a môžeme ísť na ďalšiu udalosť.

3.4 Možné vylepšenia

Počas návrhu a implementovania tohto modelu mi napadli viaceré vylepšenia, mnohé z nich už však nebol čas implementovať. Niektoré z nich predstavujú mierne zlepšenie

rýchlosti alebo presnosti algoritmov a to často iba v úzkej množine prípadov.

Manažovanie zoznamu PSP a odhad hodnoty MP

Počas implementovania algoritmu na výpočet MP som si všimol, že často je veľký počet PSP ktoré si neurón pamätá absolútne nedôležitých a ich vplyv na koncovú hodnotu MP je minimálny. Ďalší prípad bol, keď už po sčítaní malej časti PSP bolo jasné, že hodnota MP prekročí aktivačný prah a napriek tomu sa jeho hodnota počítala presne.

Skúsil som teda navrhnúť mechanizmus ako by sa PSP dali zoradiť podľa veľkosti vplyvu na hodnotu MP. Veľkosť vplyvu PSP je závislá od dvoch vecí: času (fázy v ktorej PSP je) a váhy synapsy, ktorou impulz prišiel. Je teda potrebné rozdeliť jednotlivé PSP podľa tých dvoch vlastností. Použil som na to dvojrozmerné pole, ktoré obsahovalo v každom políčku zoznam PSP. Tie boli pozaraďované do tohto poľa v jednej dimenzii podľa absolútnej hodnoty váhy synapsy, ktorá určovala ich amplitúdu. Druhá dimenzia predstavovala rozdelenie podľa času začiatku PSP. Rozdeľovalo sa podľa zvyšku po delení hodnotou, ktorá predstavovala čas ktorý je hodnota PSP pre nás zaujímavá (dostatočne sa líši od nuly). Celkovo boli teda v jednom políčku poľa tie PSP, ktoré mali podobnú absolútnu hodnotu amplitúdy a zároveň začali v podobnom čase. Mali sme teda PSP pozoskupované podľa toho ako veľmi v danom okamihu môžu či už kladne alebo záporne ovplyvniť hodnotu MP. Je samozrejmé, že v rôznych časoch je vplyv jednotlivých PSP na MP rôzny. Ale v každom časovom okamihu vieme, ako veľmi môžu jednotlivé PSP zaraďené do skupín podľa času ovplyvniť MP vzhľadom na amplitúdu akú majú. Jednoduchý spôsob ako toto využiť je vytvoriť poradie všetkých políčok dvojrozmerného poľa podľa maximálnej hodnoty, ktorou môžu ovplyvniť MP. Takéto poradie spravíme pre každý časový okamih. Je zjavné, že nakoniec bude existovať iba toľko rôznych poradí koľko je rôznych indexov v časovej osi.

Vďaka tomuto poľu vieme rozdeliť PSP na skupiny a následne ich spočítavať tak, aby sme začali s tými ktoré budú mať na MP najväčší vplyv. Tento prístup nám nepomôže, keď chceme zistiť hodnotu MP presne ale keď napríklad chceme vedieť či je hodnota nad alebo pod určitou hranicou. To môžeme skôr zistiť tak, že si do dvoch premenných

uložíme súčet negatívnych a pozitívnych amplitúd. Vždy keď pripočítame ďalší PSP, tak jeho amplitúdu odpočítame od jednej z premenných v závislosti na tom či ide o inhibičný alebo excitačný PSP. Hodnoty týchto premenných nás počas výpočtu informujú o tom o koľko sa ešte hodnota môže posunúť hore alebo dole. Ak napríklad je súčet doterajších PSP pod určitou hranicou a z premennej počítajúcej zvyšné pozitívne amplitúdy vidíme, že sa už hodnota MP nemôže posunúť hore dosť na to aby dosiahla danú hranicu. Z toho ešte pred spočítaním všetkých vieme, že hranicu už nedosiahne.

Bohužiaľ tento postup sa dal použiť iba na jednom mieste v algortime, a to vtedy keď pri udalosti typu "Koniec refraktornej fázy" zisťujeme, či je MP nad alebo pod aktivačným prahom. Na všetkých ostatných miestach potrebujeme presnú hodnotu MP a tam je pôvodný prístup rýchlejší, lebo si v ňom ušetríme zbytočné manipulácie. Napriek veľkým nádejám ktoré boli vkladané v tento prístup sa ukázalo, že v priemernom prípade neurýchlili ani ten jeden výpočet na ktorý bol použiteľný. Preto bol nakoniec z algoritmu vylúčený.

Napriek tomu je možné tento postup úspešne použiť v extrémnych prípadoch, keď odchýlka hodnoty MP je dosť vzdialená od prahu alebo inej sledovanej hranice.

3.5 Testy efektívnosti

Výpočtová efektívnosť celého procesu je založená na úspešnosti iteračného algoritmu na odhad času prekročenia aktivačného prahu. Všetky ostatné časti spotrebujú v prepočte na jeden odsimulovaný impulz iba malé skoro konštantné množstvo času. Preto môžeme pre zjednodušenie považovať efektívnosť algoritmu priamo úmernú rýchlosti iteračného algoritmu.

Na to aby som zistil ako úspešný je algortimus pri odhadovaní času vzniku impulzu, spravil som sériu testov, ktoré porovnávajú rýchlosť nie len pre rôzne parametre modelu ale aj rôznu požadovanú presnosť.

–	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
0.10 <i>ms</i>	4.44	5.24	6.05	7.03	7.61
0.15 <i>ms</i>	–	4.64	5.60	6.47	7.35
0.20 <i>ms</i>	3.36	4.49	5.24	6.25	7.12
0.25 <i>ms</i>	–	4.41	5.28	6.39	7.00
0.30 <i>ms</i>	3.06	4.35	5.41	6.40	7.29
0.40 <i>ms</i>	3.12	4.48	5.41	6.47	7.49
0.60 <i>ms</i>	3.12	4.67	5.81	6.96	8.05
1.00 <i>ms</i>	3.25	4.82	6.23	7.82	9.43

V prvej tabuľke môžeme vidieť priemerné množstvo iterácií, ktoré algoritmus potreboval na to aby pri zvolenej presnosti našiel časový bod v ktorom MP prekročí aktivačný prah. Každý riadok predstavuje pokusy pre nejakú hodnotu parametra t_d , ktorý som použil v hodnote od 0.1 ms do 1.0 ms. Každý stĺpec zase predstavuje určitú zvolenú presnosť. Vyskúšané presnosti boli v rozhraní od 10^{-4} sekundy do 10^{-8} sekundy. Hodnoty boli vypočítané z 10000 odsimulovaných impulzov z ktorých si každý vyžadoval použitie iteratívneho algoritmu a v konečnom dôsledku mohol spôsobiť prekročenie aktivačného prahu. Teda impulzy ktorých odsimulovanie skončilo v niektorom optimalizačnom teste, lebo bolo jasné že nespôsobia aktiváciu neurónu, sa do výpočtu týchto hodnôt nedostali. Taktiež sa do týchto hodnôt nedostali impulzy, ktoré síce použili pri predpovedi správania MP daný algoritmus, ale v konečnom dôsledku bol výsledok negatívny, čo sa možnosti dosiahnutia aktivačného prahu týka. Tieto impulzy boli základom na nasledujúcu tabuľku.

–	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
0.10 <i>ms</i>	8.89	8.89	8.92	8.79	8.83
0.15 <i>ms</i>	--	5.98	5.93	5.93	5.95
0.20 <i>ms</i>	4.93	4.95	4.99	4.93	4.98
0.25 <i>ms</i>	--	4.06	4.24	4.09	4.07
0.30 <i>ms</i>	3.10	3.18	3.33	3.32	3.57
0.40 <i>ms</i>	3.06	3.24	3.30	3.34	3.39
0.60 <i>ms</i>	2.71	2.95	3.34	2.92	2.95
1.00 <i>ms</i>	2.90	3.34	3.26	3.47	4.00

V druhej tabuľke sú uvedené hodnoty priemerného množstva iterácií potrebných na to aby algoritmus odhalil, že nedôjde k prekročeniu aktivačného prahu. V riadkoch a stĺpcoch je tabuľka organizovaná rovnako ako prvá tabuľka. Hodnoty boli teda získané z 10000 impulzov, ktoré použili iteračný algoritmus a ten zistil, že nemôžu spôsobiť prekročenie aktivačného prahu.

Všetky tieto hodnoty boli získané za použitia siete 500 neurónov (400 excitačných a 100 inhibičných), náhodne pospájaných. Spojenia boli náhodné s pravdepodobnosťou 10%. Pre hodnoty PSP boli použité hodnoty parametrov $t_1 = 0.5$ a $t_2 = 2.0$. Ako vstup do siete bol použitý šum, teda náhodná aktivita prichádzajúca do každého neurónu.

Samozrejme hodnoty nie sú veľmi smerodajné, lebo sa môžu veľmi líšiť od prípadu k prípadu. A sú náchylné na zmenu jednotlivých parametrov modelu. Testy hlavne slúžili na demonštráciu toho, že iteračný algoritmus je efektívny. V prvej tabuľke vidieť, že rozdiel medzi dvoma stĺpcami je približne jedna. Čiže na spresnenie predpovede o jedno desatinné miesto potrebujeme ani nie jednu celú ďalšiu iteráciu. Keď uvažujeme impulzy, ktoré nespôsobia prekročenie aktivačného prahu, tak vplyv zvolenej presnosti na rýchlosť algoritmu je minimálny.

Napriek efektívnosti tohto algoritmu, veľmi málo impulzov ktorých vplyv sa vyhodnocuje potrebuje na predpoveď správania tento náročný nástroj. V tomto konkrétnom pokuse, čiže v sieti ktorá približne zodpovedá hustotou a aktivitou reálnej neurónovej sieti

až na 93% impulzov stačil jeden z jednoduchých testov. Tieto testy, ako som už spomenul vyššie v kapitole o algoritme, slúžia práve na to aby sme jednoducho určili či je potrebné tento náročný algoritmus použiť, alebo či sme schopní aj bez neho určiť či môže dôjsť k prekročeniu prahu. Samozrejme s nižšou aktivitou siete môže toto percento ešte viac vzrásť.

Ďalší fakt s ktorým treba pri zväčšovaní presnosti rátať, je ten, že pre vybranú presnosť treba prispôbiť predpočítané dáta. To znamená, že ak zvolíme extrémnu presnosť napr. 10^{-9} sekundy a dĺžka PSP je 5 ms, tak je potrebné predpočítať 5 000 000 hodnôt. A to je iba pre hodnoty PSP, ďalších 5 000 000 je to pre derivácie PSP.

3.6 Paralelizácia

Základným problémom všetkých systémov simulujúcich neurónovú sieť je, že sú založené na úplne odlišnej hardvérovej architektúre ako samotný mozog. Simulovať výpočtové schopnosti miliónov alebo miliard neurónov pomocou jedného výpočtového zariadenia je skoro nemožné. Keď do boja zapojíme viac procesorov tak môžeme túto nepriaznivú situáciu zlepšiť (viď. [9]), ale pri naozaj veľkých sieťach nám ani toto nepomôže. Viacprocesorové architektúry majú mnohé ohraničenia a zvyšovanie počtu procesorov je problematické. Ale sú samozrejme aj iné prístupy z ktorých je každý použiteľný inde.

Vo všeobecnosti máme tieto prístupy ako paralelizovať výpočet:

- Časová paralelizácia - Ak nás nezaujíma samotné dianie v sieti a našim cieľom je iba naučiť sieť niečo, tak môžeme tento prístup použiť. Majme sieť ktorú chceme niečo naučiť a máme k tomu 1000 vstupov na ktorých ju treba natrénovať. Ak máme povedzme 5 procesorov, tak podhodíme každému 200 vstupov a natrénujeme sieť na každom z nich oddelene. Nakoniec sčítame zmeny ku ktorým v sieti došlo počas učenia na jednotlivých výpočtových strojoch. V princípe ide teda o rozdelenie vstupov medzi výpočtové jednotky. Tento prístup sa používa pri tzv. Batch algoritme na tréning samoorganizujúcich sietí (Batch SOM) (viď. [9]). Tento prístup nie je náročný na medziprocesorovú komunikáciu, čiže je ľahko realizovateľný na viacprocesorových architektúrach alebo na sieti (LAN).

- Priestorová paralelizácia - Pri priestorovej paralelizácii sa snažíme rozdeliť sieť medzi výpočtové jednotky tak, aby každá jednotka mala na starosť nejakú súvislú skupinu neurónov. Tento prístup je však značne problematický. Ak nám simulácia prebieha v cykloch, tak ako fungujú skoro všetky modely, tak vznikne problém v komunikácii. Po cykle, keď sa prepočíta správanie každého neurónu v jednotlivých výpočtových strojoch, vznikne naraz veľké množstvo dát, ktoré treba presunúť. Tieto dáta predstavujú impulzy, ktoré prechádzajú medzi neurónmi patriacimi do rôznych výpočtových jednotiek. Keď uvažujeme že prenos informácie je časovo náročná operácia a musíme ju vykonať pre mnoho dvojíc nezávislých jednotiek, tak strácame podstatnú časť výpočtového potenciálu, lebo procesory musia čakať, kým skončí prenos. Práve pre tento problém sa tento prístup len zriedkavo používa.

Jednou z hlavných výhod môjho modelu je, že je vhodný pre priestorovú paralelizáciu. Jeho prioritným zameraním sú veľké a biologicky realistické siete, na ktorých simuláciu treba v praxi použiť viacero výpočtových zariadení naraz. Model je možno pomalší a nie veľmi vhodný pre malé a husté siete, ale v kontexte biologicky relevantných sietí uvidíme jeho výhody.

Zoberme si ako príklad sieť, ktorú použil Izhikevich v [5]. Táto sieť je, aspoň podľa článku, biologicky relevantná. Jednalo sa o sieť 100 000 neurónov, ktoré boli rozmiestnené rovnomerne na povrchu gule s polomerom $8mm$. Neuróny boli náhodne na 80% excitačné a na 20% inhibičné. Každý excitačný neurón bol spojený s neurónmi v okolí $1.5mm$ s pravdepodobnosťou 9% a viedol z neho ešte axón. Tento Axón mal dĺžku $12mm$ a viedol do náhodnej vzdialenejšej časti gule, kde sa náhodne napojil na neuróny v okruhu $0.5mm$. Podobne to fungovalo aj s inhibičnými neurónmi, len polomer pripojení bol u nich trochu odlišný. Dôležitá časť je ale tá, že oneskorenia spojení sú priamo úmerné ich vzdialenosti. Z toho dostávame, že blízke neuróny su spojené s malým oneskorením a môžu mať aj spojenie do vzdialenejších častí, to však s väčším oneskorením. Keďže vieme akú architektúru má biologická sieť, tak môžeme navrhnúť spôsob ako ju úspešne paralelizovať.

Naším cieľom je rozdeliť celú simulovanú sieť tak, aby každý neurón patril práve do jednej skupiny. A aby sme minimalizovali potrebu komunikácie medzi skupinami a zároveň maximalizovali dĺžku oneskorenia spojení medzi rôznymi skupinami. Minima-

lizácia potrebnej komunikácie znamená vytvoriť skupiny tak, aby bolo medzi jednotlivými rôznymi skupinami čo najmenej spojení. Táto úloha je však dosť náročná a snaha o úplnu minimalizáciu je zväčša aj tak zbytočná. Napriek tomu sa z architektúry siete dá často ľahko vyčítať optimálna cesta ako dané neuróny rozdeliť tak, aby sme náš cieľ relatívne splnili. Význam potreby redukcie medziskupinovej komunikácie je jasný, keďže komunikácia je hlavným problémom tohto prístupu pri paralelizácii, je prirodzené sa snažiť ju minimalizovať. Oveľa dôležitejšia je však snaha maximalizovať oneskorenie najrýchlejšieho spojenia medzi neurónmi. Dôležitosť oneskorenia ukážem na ďalšom príklade.

Majme skupinu neurónov A a skupinu B. Najrýchlejšie spojenie zo skupiny A do skupiny B má oneskorenie povedzme $20ms$ a opačným smerom $25ms$. Keď dôjde k synchronizácii skupín (skupiny si vymenia veškeré informácie o impulzoch, ktoré cestovali z jednej skupiny do druhej), môžu skupiny chvíľku operovať samostatne. Ak sa už vie o nejakej impulze čo má prejsť zo skupiny A do B v čase synchronizácie, tak sa údaj o tomto impulze preniesie. Ak sa o tomto impulze nevie, to znamená že ešte nedošlo k jeho vzniku v neuróne a vieme že najskôr môže doraziť do skupiny B až o $20ms$. Lebo keď vznikne (v čase po synchronizácii) tak ho ešte čaká cesta nejakým spojením, čiže určite neovplyvní nijakým spôsobom simuláciu skupiny B skôr ako o $20ms$. V tomto prípade sa môže skupina A simulovať nezávisle od skupiny B až $25ms$ a skupina B zase $20ms$. Táto vlastnosť nám umožňuje simulovať obrovské množstvo skupín, bez potreby globálnej synchronizácie. Každá dvojica skupín sa musí synchronizovať jedným smerom iba tak často ako aké je najrýchlejšie spojenie medzi nimi.

Otázne je už iba ako sa nám podarí rozdeliť sieť do skupín. Z vlastností reálnych biologických sietí, ktoré sú relatívne riedke môžeme usúdiť, že by tento prístup mohol priniesť určitý pokrok v možnostiach simulácie naozaj veľkých sietí. Zatiaľ určite nie v reálnom čase (simulačný čas uteká rovnako rýchlo ako reálny), ale s určitým spomalením by sme mali byť schopní simulovať sieť takú veľkú koľko výpočtového hardvéru sme schopní spojiť pomocou siete (LAN).

4 Samoorganizujúca mapa pomocou JASTAP

4.1 Impulzné samoorganizujúce mapy

V posledných rokoch sa viacero výskumných prác venovalo využitiu impulzných modelov neurónov v samoorganizácii. Taktiež sa podarilo dosiahnuť viaceré výsledky v tejto oblasti. Impulzné neuróny sa podarilo úspešne použiť pri samoorganizácii jednoduchej mapy, ako môžeme vidieť v prácach [10] a [13]. A podobne na poli clusterizácie boli dosiahnuté pokroky, keď bol algoritmus založený na impulzných neurónoch úspešne aplikovaný na vysokorozmerné dáta (viď [11] a [12]).

4.2 Samoorganizácia

Princíp samoorganizácie je založený na klasickom koncepte navrhnutom Kohonenom (viď [14]). Máme homogénnu a pravidelnú sieť (napr. štvorcovú mriežku) a snažíme sa jednotlivé prvky tejto siete umiestniť do zvyčajne viacrozmerného priestoru vstupov tak, aby sa každý prvok siete stal reprezentantom nejakej podmnožiny množiny vstupov. Dôležité sú však susedské vzťahy prvkov v sieti, lebo tie po úspešnom namapovaní ukazujú vzťahy (určitú podobnosť a rozdielnosť) jednotlivých podmnožín množiny vstupov, ku ktorým boli priradené.

Základný mechanizmus samoorganizácie ako bol navrhnutý Kohonenom je založený na hľadaní víťazov. Pre každý prvok vstupu sa nájde víťaz, čiže prvok (neurón) siete, ktorý sa určitým spôsobom najviac podobá danému vstupu. Podobnosť môžeme chápať napríklad ako vzdialenosť v priestore alebo ako skalárny súčin vektorov ak ide o normalizované vektory. Keď tohto víťaza nájdeme, tak jeho a aj jeho okolie (určitý okruh jeho susedných neurónov) posunieme k danému vstupu, aby nabudúce boli podobnému vstupu ešte bližšie. Aby sme zabránili tomu aby sa nám sieť stiahla do jedného bodu, musíme neuróny za týmto okruhom susedov odtlačiť zase ďalej, čím dosiahneme že nabudúce budú podobnému vstupu ešte ďalej. Na riadenie toho, ktoré neuróny sa priblížia vstupu a ktoré vzdialia sa používa tzv. funkcia mexického klobúka (pre viac detailov pozri [14]).

Teraz sa pozrime na to ako sa dá podobný mechanizmus simulovať pomocou impulzných

neurónov. Postup bude rovnaký ako v prácach [10] a [13]. Každý neurón bude mať toľko vstupných synaptických spojení koľko má vstupný priestor dimenzií. Hodnotu jednotlivých zložiek vstupného vektora budeme reprezentovať ako oneskorenie jediného impulzu, ktorý príde zodpovedajúcim synaptickým spojením. Čiže pri vkladaní daného jedného vstupu bude vytvorený impulz pre každý jeden neurón a každé jeho vstupné synaptické spojenie (teraz nehovoríme o laterálnych spojeniach). Predpokladajme, že hodnoty zložiek vstupných vektorov sú hodnoty z intervalu $\langle 0, 1 \rangle$ a vektory sú normalizované. Potom oneskorenie impulzov reprezentujúcich jednotlivé hodnoty budú:

$$t_j = (a_{jmax} - a_j) \cdot t_{in}$$

Kde t_j je oneskorenie, ktoré impulz bude mať. a_{jmax} je maximálna hodnota zložky z daného vstupného vektora a a_j je hodnota danej zložky, ktorú týmto impulzom chceme reprezentovať. Čas t_{in} určuje škálu oneskorenia, je to parameter a jeho optimálna hodnota bude popísaná nižšie.

Základný predpoklad fungovania celého mechanizmu je, že prvá časť funkcie PSP je skoro lineárna. Keď sa pozrieme na tvar funkcie PSP (obr. 8) ktorú používame tak vidieť, že kým funkcia nedosiahne svoje maximum, tak jej priebeh je približne lineárny. Pozrime sa teda na reakciu jednotlivých neurónov na takýto vstup. Ak v nejakom neuróne vznikne impulz, tak to bude najskôr v čase $t_{imp} = \min\{t \mid \sum_j PSP(t - t_j) \geq \theta\}$ (θ je hodnota aktivačného prahu). Potom za predpokladu, že ku prekročeniu aktivačného prahu vo víťaznom neuróne dôjde skôr ako niektorý PSP dosiahne svoje maximum, môžeme zjednodušiť jednotlivé PSP na lineárnu funkciu. Teda čas kedy dôjde k prekročeniu AP sa dá približne vypočítať: $t_{imp} = (\theta - a \sum_j t_j w_j) / a$. Kde θ je hodnota AP, a je sklon lineárnej funkcie ktorou sme zjednodušili začiatok funkcie PSP, t_j je oneskorenie impulzu v j -tej synapse a w_j je váha j -tej synapsy. Keďže hodnota a je pre nás konštanta závislá od tvaru PSP (ktorý sa nemení), tak vidieť že čas prekročenia aktivačného prahu je priamo závislý od skalárneho súčinu vektorov w' a $-t'$. Kde w' je vektor váh jednotlivých synáps neurónu a t' je vektor oneskorenia jednotlivých vstupných impulzov. Tu vidieť paralelu s Kohonovým prístupom, kde víťaz bol neurón s maximálnym skalárnym vektorom. Rovnako ako aj v tomto algoritme bude víťaz ten, ktorý má najväčší skalárny súčin vektorov w' a $-t'$.

Namiesto toho aby sme použili funkciu "mexického klobúka" až priamo v učení použijeme ju tak, že podľa nej nastavíme váhy laterálnych spojení medzi neurónmi. Každý neurón má laterálne spojenia s blízkymi neurónmi kladné a s vzdialenejšími záporné. Význam bude taký, že keď sa nájde víťaz tak impulz ktorý on vygeneruje sa prenesie cez laterálne spojenia a následne pomôže okolitým neurónom dosiahnuť AP skôr a vzdialenejšie inhibuje, čiže tie budú aktivované neskôr. Toto dáva samozrejme až zmysel v kontexte fungovania samotného učenia. Modifikácie učenia menia váhy vstupných laterálnych spojení tak, že priblížia vektor váh bližšie k vektoru daného vstupu pre neuróny ktoré reagovali rýchlo a tie čo nezareagovali dostatočne rýchlo budú pozmenené opačne. Modifikačné pravidlo vyzerá takto:

$$\Delta w_{ij} = \eta(t) \cdot \frac{T_{out} - t'_i}{T_{out}} \cdot (a_j - w_{ij})$$

Kde w_{ij} je váha j-tej vstupnej synapsy i-teho neurónu, η je časový priebeh intenzity učenia, T_{out} je časová konštanta určujúca hranicu, ktoré neuróny budú priblížené k vstupu a ktoré oddialené. Hodnota t'_i určuje čas vzniku impulzu v i-tom neuróne a a_j je j-ta vstupná hodnota.

4.3 IRIS

Pre začiatkový test som sa rozhodol skúsiť namapovať dvojrozmernú sieť JASTAP neurónov na databázu IRIS. Ide o databázu kvetov, z ktorých každý je reprezentovaný vektorom s reálnymi hodnotami. Pre túto alternatívu som sa rozhodol preto, lebo o kategorizáciu kvetov IRIS sa už pokúšalo mnoho modelov a ich rôzna úspešnosť je dobre zdokumentovaná.

Presnejšie IRIS je databáza 3 rodov kosatcov. Každý exemplár je popísaný 4 číselnými hodnotami, ktoré vyjadrujú: šírku a dĺžku kališného lístka a šírku a dĺžku okvetných lístkov. Veľa kategorizačných algoritmov tu dosahuje vysokú úspešnosť napriek tomu, že nie všetky exempláre sú jednoznačne zatrieditelné. To je spôsobené tým, že dva rody sa v identifikačných hodnotách prekrývajú. Najlepšie kategorizačné algoritmy tu aj napriek tomu dosahujú presnosť až do 95%.

V tomto teste som použil databázu IRIS obsahujúcu 150 rôznych jedincov. Ako presne vyzerá rozloženie vektorov reprezentujúcich jednotlivé exempláre v prístore je možné vidieť v prílohe A.

4.4 Sieť

Inšpirovaný predošlými prácami, kde boli impulzné modely neurónov úspešne použité, navrhol som malú dvojrozmernú sieť neurónov. Išlo o sieť 8x8 neurónov JASTAP, ktoré boli laterálne spojené každý s každým. Všetky laterálne spojenia mali oneskorenie lineárne závislé od vzdialenosti neurónov v sieti. Každý neurón mal synaptické spojenie predstavujúce jednu dimenziu vstupného vektora pre všetky dimenzie vstupu. To znamená, že pomocou týchto synaptických spojení dostal vstupný signál každý neurón siete. Potom ako sieť dostala vstup sa počkalo na reakciu neurónov a podľa určených pravidiel nastalo učenie. Pozrime sa na jednotlivé aspekty siete teraz podrobnejšie.

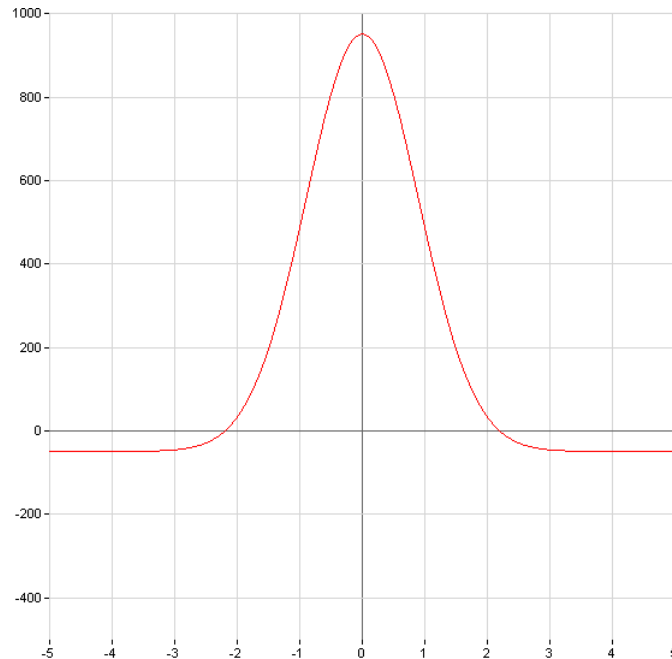
4.4.1 Laterálne spojenia

Samotný model samoorganizujúcej mapy vychádza zo známeho konceptu Kohonena (viď [14]). Vzhľadom na tento koncept plnia laterálne spojenia úlohu väzieb medzi blízkymi neurónmi. Ich úlohou je držať koncové umiestnenie (význam) susedných neurónov blízko seba aby sa pri namapovaní siete na priestor vstupov zachovala štruktúra sieťovej mriežky. V tomto prípade to budú robiť pomocou toho, že ak sa nájde víťazný neurón (ten ktorý na vstup zareaguje vznikom impulzu ako prvý) tak tento neurón pomocou excitačných laterálnych spojení excituje blízke neuróny, čím zrýchľuje ich reakciu na vstup. Tým že okolité neuróny budú reagovať skôr, tak v učiacom algortime budú modifikované podobne ako víťazný neurón. Podobne pre laterálne spojenia vzdialených neurónov bude efekt presne opačný, lebo tie sú od určitej hranice inhibičné.

Na váhy laterálnych spojení sa použila myšlienka mexického klobúka (viď [14]). Ide o pravidlo hovoriace, že víťaz má blízke neuróny posilniť (pritiahnuť k sebe) a vzdialenejšie oslabiť (odtlačiť od seba).

$$w_{ij} = e^{\frac{-d_{ij}^2}{a}} \cdot b - c(t) \quad (2)$$

Obrázok 9: Graf funkcie mexického klobúka



Graf funkcie mexického klobúka, práve pre parametre v 250 kroku učenia.

Graf funkcie vidieť na (obr. 9). V rovnici d je vzdialenosť neurónov ($d = 1$ pre vedľajšie neuróny), a , b a $c(t)$ sú parametre, kde c sa môže s časom meniť. Po mnohých pokusoch, počas ktorých bolo veľkým problémom nastaviť tieto parametre som došiel k týmto hodnotám: $a = 1.6$, $b = 1000.0$, $c(0) = 0.0$, $c(t) = \text{if } c(t - 1) < 800 \text{ then } c(t - 1) + 0.2 \text{ else } c(t - 1)$.

Oneskorenia jednotlivých spojení medzi neurónmi boli podobne ako v prácach Izhikevicha ([3], [4] a [5]) lineárne závislé od vzdialenosti neurónov. Čiže na výpočet oneskorenia bol použitý nasledovný vzťah.

$$t_{sd} = g \cdot d_{vzdialenost}$$

Kde t_{sd} je výsledné oneskorenie, $d_{vzdialenost}$ je vzdialenosť neurónov v sieti a g je konštanta (parameter modelu celkovo), ktorou môžeme nastaviť dĺžku oneskorenia. Toto oneskorenie môžeme konečne nastaviť tak, aby nám vyhovoval jeho vplyv na správanie siete. Samotné oneskorenia nemajú v pôvodnom koncepte SOM žiadnu paralelu, ale ich

použitie je vzhľadom na biologickú stavbu neurónových sietí prirodzené. Preto som ich vplyv na správanie siete preskúmal.

Ak sa v modeli použil parameter na dĺžku oneskorení s hodnotou $g = 0$, alebo s hodnotou veľmi nízkou, tak sme dostali správanie ktoré zodpovedá konceptu SOM z prác [14] a výsledný efekt bol porovnateľný s výsledkom z prác [10] a [13]. Keď sa aktivoval prvý neurón a vytvoril impulz, tak tento impulz sa okamžite rozšíril po celej sieti a spôsobil tzv. efekt mexického klobúka, ktorý sa používa v štandardných SOM. Ten pridá aktivitu blízkym a odoberie vzialeným neurónom bez akejkoľvek straty efektu. Ak sa g použilo s väčšou hodnotou, tak impulzy z víťazného neurónu a potom ďalších neurónov, ktoré boli aktivované, dorazili s oneskorením. Čo spôsobilo hlavne to, že ich efekt na vzdialenejšie neuróny bol menší. To znamená, že vplyv samotných oneskorení bol iba ten, že znižovali efekt laterálnych spojení pre vzdialenejšie neuróny, čím iba menili tvar funkcie mexického klobúka. Čo nie je vôbec potrebné, lebo túto funciu si môžeme nastaviť tak ako je pre daný prípad optimálne. Na základe týchto úvah bolo teda oneskorenie použité v laterálnych spojeniach nulové.

4.4.2 Normalizácia vstupu

Na to aby sme databázu IRIS mohli použiť pri tomto spôsobe učenia bolo nutné urobiť niekoľko normalizačných krokov na vstupných dátach. Tak ako je uvedené v [14] pre hľadanie víťazného neurónu riadeného skalárnym súčinom je nutné mať vstupné vektory rovnakej dĺžky a podľa možnosti s čo najväčšími rozdielmi v smere.

Ako prvý normalizačný krok na všetkých vstupných vektoroch bola tzv. min-max normalizácia (min-max normalization, viď [15]). Úlohou tejto normalizácie je posunúť hodnoty vstupov tak, aby efektívne pokrývali priestor v ktorom sme schopní operovať. V tomto prípade išlo o posun hodnôt všetkých zložiek jednotlivých vstupných vektorov na hodnoty z intervalu $\langle 0, 1 \rangle$. (Vstupné dáta pred a po celkovej normalizácii môžete vidieť v prílohe A). V skutočnosti bolo iba na všetky zložky vstupných vektorov aplikované nasledovné pravidlo (vektor $v = (v_1, \dots, v_n)$):

$$v_i := \frac{(v_i - v_{i_{min}})}{v_{i_{max}} - v_{i_{min}}}$$

kde $v_{i_{min}}$ ($v_{i_{max}}$) je minimum (maximum) z i -tych zložiek všetkých vstupných vektorov.

Druhý krok bola úprava vektorov tak, aby mali rovnakú veľkosť. Na tento účel som pridal k vektorom ďalší rozmer, ktorého hodnota bola vždy taká, aby všetky vektory mali rovnakú dĺžku. Čiže pre vektor $v = (v_1, \dots, v_n)$ som vyrobil ešte jednu zložku v_{n+1} , pre ktorú platí.

$$v_{n+1} = \sqrt{m_{max} - \sqrt{\sum_{i=1}^n v_i^2}}$$

kde

$$m_{max} = \max\left\{\sqrt{\sum_{i=1}^n v_i^2} \mid \forall v = (v_1, \dots, v_n)\right\}$$

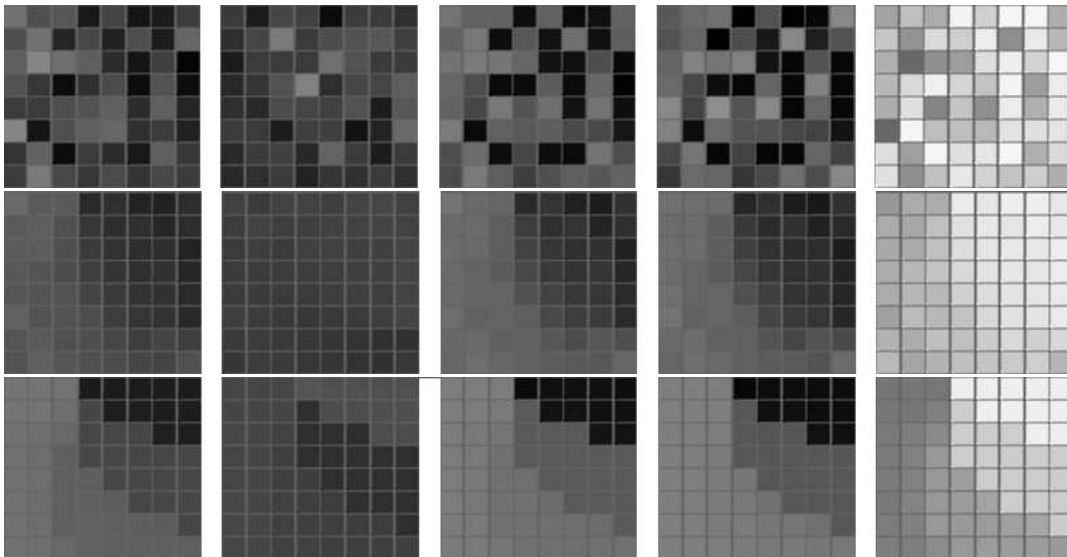
Po tomto poslednom kroku boli už vstupné dáta vhodné pre použitie v učení založenom na skalárnych súčinoch.

4.4.3 Pribeh učenia

Samotné učenie prebiehalo v krokoch. Pred prvým krokom boli všetky váhy vstupných synapsií nastavené na náhodné kladné hodnoty tak, aby vektor váh pre každý neurón mal dĺžku jedna. Na začiatku každého kroku bola sieť v pokoji (resetovaná), to znamená všetky neuróny boli v pokojovom stave a synaptickými spojeniami necestovali žiadne impulzy. Do takejto pokojnej siete boli dodané vstupy, ktoré prinútili niektoré neuróny vystreliť impulz a tým samozrejme ovplyvniť aj okolité neuróny. Čas aktivovania jednotlivých neurónov bol zaznamenaný a po uplynutí určitého krátkeho časového okna, bola sieť znova resetovaná a bolo aplikované učenie podľa pravidla uvedeného vyššie za pomoci údajov o čase aktivácie jednotlivých neurónov (Tento princíp učenia má aj biologické opodstatnenie, pre podrobnosti pozri [10] a [13]). V databáze bolo 150 vstupov, ktoré sa náhodne striedali v jednotlivých krokoch učiaceho algoritmu. Rýchlosť učenia bola určená nasledujúcim vzťahom $\eta(t) = 0.2 \cdot 0.9998^t$, kde t je poradové číslo daného kroku učenia.

Po stovke krokov bolo už jasne vidieť formovanie siete a definitívne sa sieť ustálila

Obrázok 10: Učenie



Zmeny vstupných váh následkom učenia. Každý riadok predstavuje stav siete v určitom okamihu. Každý z piatich štvorcov v riadku predstavuje váhy synapsí jednotlivých neurónov zodpovedajúcich jednej z piatich dimenzií vstupu. Čierna farba reprezentuje hodnotu 0 a biela hodnotu 1. Prvý riadok je rozloženie váh na začiatku učenia, druhý po 200 krokoch a tretí po 2500 krokoch.

približne po 2000 krokoch. Na (obr. 10) vidieť vývoj váh jednotlivých vstupných synapsí, ktoré sa postupne prispôsobujú charakteru vstupných vektorov. Postupne sa vytvorili skupiny neurónov, ktoré majú podobné váhy a reagujú navzájom skoro rovnako na jednotlivé vstupy. Ako na (obr. 10) vidieť, po 200 krokoch sa už skupiny začínajú formovať, ale hranice medzi nimi ešte nie sú jasné. Po 2500 krokoch sú už jednotlivé skupiny jasne formované a navzájom rozlíšiteľné. Po 2500 nastal stav, že pri každom zo stupov každá zo skupín buď:

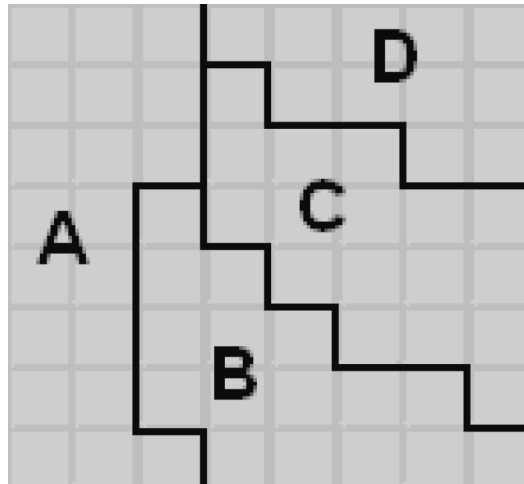
- Sa aktivovala celá. Tým myslím, že každý jej neurón vygeneroval impulz. Pritom vo všetkých jej neurónoch došlo k prekročeniu aktivačného prahu skoro naraz.
- Zostala neaktívna. Čiže v ani jednom jej neuróne nedošlo k prekročeniu aktivačného prahu pred skončením časového intervalu určeného pre daný krok učenia.

Toto správanie odlišuje tento prístup od Kohonenovej SOM, ktorá mapuje sieť rovnomerne a nevznikajú takéto jasné hranice, iba plynulé prechody. Táto odlišnosť je spôsobená tým ako sa používa funkcia mexického klobúka pri učení. Ja ju nepoužívam priamo, ale iba sprostredkované cez váhy laterálnych spojení, preto je jej efekt dosť obmedzený. Túto vlastnosť však môžeme náležite využiť. To že jednotlivé skupiny, ktoré v sieti vznikli sú ľahko rozlíšiteľné a buď sú alebo nie sú aktivované jednotlivými vstupmi môžeme skúsiť použiť na ich kategorizáciu. Každé skupine neurónov priradíme jednu triedu vstupov a keď dôjde k aktivácii danej skupiny, tak to budeme považovať ako odpoveď siete, že priradila vstup do danej skupiny.

Ako výsledok učenia vznikli 4 jasne odlišiteľné skupiny (obr. 11). A ako tieto skupiny reagovali na vstupy? Ako som už skôr spomínal, vstupné dáta obsahovali 150 rôznych exemplárov rastlín. Išlo o jedince 3 rôznych rodov (50 exemplárov z každého rodu), ktoré by sieť mohla byť schopná rozlíšiť. Z týchto 150 rôznych vstupov sieť reagovala na 141 z nich aktiváciou jednej skupiny a na 9 aktiváciou 2 skupín. Presné reakcie na jednotlivé rody boli nasledovné:

- Prvá trieda (Iris-setosa) pozostávajúca z 50 exemplárov aktivovala v 96% prípadov skupinu **D** a v 4% prípadov skupiny **C** aj **D**. Z toho vidieť, že Iris-setosa vždy aktivovala skupinu **D** a niektoré exempláre aktivovali aj skupinu **C**.

Obrázok 11: Skupiny



Skupiny ktoré vznikli ako následok učenia.

- Druhá trieda (*Iris-versicolor*, tiež 50 vzoriek) aktivovala v 80% prípadov skupinu **C**, v 4% prípadov skupinu **D**, v 8% prípadov skupinu **B** a v 8% prípadov skupiny **C** a **D**.
- Tretia trieda (*Iris-virginica*, tiež 50 vzoriek) aktivovala v 46% prípadov skupinu **A**, v 14% prípadov skupinu **B**, v 34% prípadov skupinu **C** a v 6% prípadov skupiny **A** a **C**.

K zmeraniu úspešnosti kategorizácie sa teraz môžeme postaviť viacerými spôsobmi. Keďže učenie prebiehalo bez učiteľa, tak musíme sami usúdiť ako budeme chápať jednotlivé odpovede. Prvá možnosť je považovať za správne iba reakcie, v ktorých bola aktivovaná iba jedna skupina. V tomto prístupe by sme aktivácie jednotlivých skupín mohli chápať takto:

- **Skupina A:** *Iris-virginica*
- **Skupina B:** *Iris-virginica*
- **Skupina C:** *Iris-versicolor*
- **Skupina D:** *Iris-setosa*

Pri rozdelení a predpoklade, že všetky reakcie aktivujúce viac ako jednu skupinu sú nesprávne dosiahneme 118 správnych odpovedí zo 150, čo predstavuje **úspešnosť 78,66 %**. Keď sa však zamyslíme nad tým, že aj aktivácia viacerých skupín môže byť korektná odpoveď, môžeme skúsiť zatriediť aj tieto prípady a to nasledovne (— znamená: taký prípad nenastal):

- Skupiny A, B: —
- Skupiny A, C: Iris-virginica
- Skupiny A, D: —
- Skupiny B, C: —
- Skupiny B, D: —
- Skupiny C, D: Iris-versicolor

Týmto dodatočným prerozdelením dosiahneme 125 úspešných odpovedí zo 150, čo predstavuje **úspešnosť 83,33 %**.

Hlavnou prekážkou správneho naučenia bola zjavne skupina C, ktorá sa ťažko interpretovala kvôli tomu, že reagovala na veľa exemplárov z dvoch rôznych skupín. Tento fakt je prirodzený, lebo podobnosť týchto dvoch rodov (Iris-versicolor a Iris-setosa) je ľahko viditeľná (viď príloha A). Napriek tomu je úspešnosť tohto prístupu v porovnaní s ostatnými kategorizačnými algoritmi podstatne horšia (ako príklad úspešného riešenia kategorizácie IRIS databázy viď práce, kde sa úspešnosť pohybuje nad 90%:[16], [17] a [18]).

4.4.4 Záver

Práca bola inšpirovaná prístupom, ktorý bol použitý v prácach [13] a [10]. Na rozdiel od týchto prác bola však použitá komplexnejšia a už ťažko nastaviteľná sieť ako aj zložitejšie vstupné dáta. Tento výsledok ukazuje, že sa impulzné neuróny dajú použiť ako alternatíva ku Kohonenovej SOM, je však oveľa obtiažnejšie nastaviť daný model pre určené vstupné dáta a dosiahnuť dobré výsledky. Hlavným cieľom tohto pokusu však nebolo

ukázať možnosti implementácie samoorganizujúcich máp pomocou impulzných neurónov, ale ukázať, že simulácia pomocou udalostí funguje rovnako dobre ako klasický prístup k modelovaniu impulzných neurónov. Aj keď sieť použitá v tomto teste nemá vôbec biologic-ky relevantné parametre (čiže môj model nie je veľmi vhodný na simuláciu takejto siete), test ukázal, že mnou navrhnutý prístup funguje aj v takejto aplikácii dostatočujúco.

5 Záver a budúci výskum

Navrhol som modifikácie impulzného modelu neurónu JASTAP, ktoré boli potrebné pre možnosť simulácie tohto modelu pomocou udalostí. Navrhol som samotný mechanizmus simulácie pomocou udalostí, ktorý umožňuje simulovať s veľkou presnosťou bez značného spomalenia simulácie. Navrhol som ďalšie modifikácie, ktoré by mohli algoritmus vylepšiť ak by bol použitý pre model s inými možnosťami. Taktiež som naznačil výhody a nevýhody tohto prístupu a ukázal oblasti, kde by sa tento mechanizmus simulácie dal úspešne použiť. Nakoniec som výpočtový model aplikoval na reálny problém, kde sa ukázali jeho parametre a vhodnosť ako aj nevhodnosť jeho použitia v určitých situáciách.

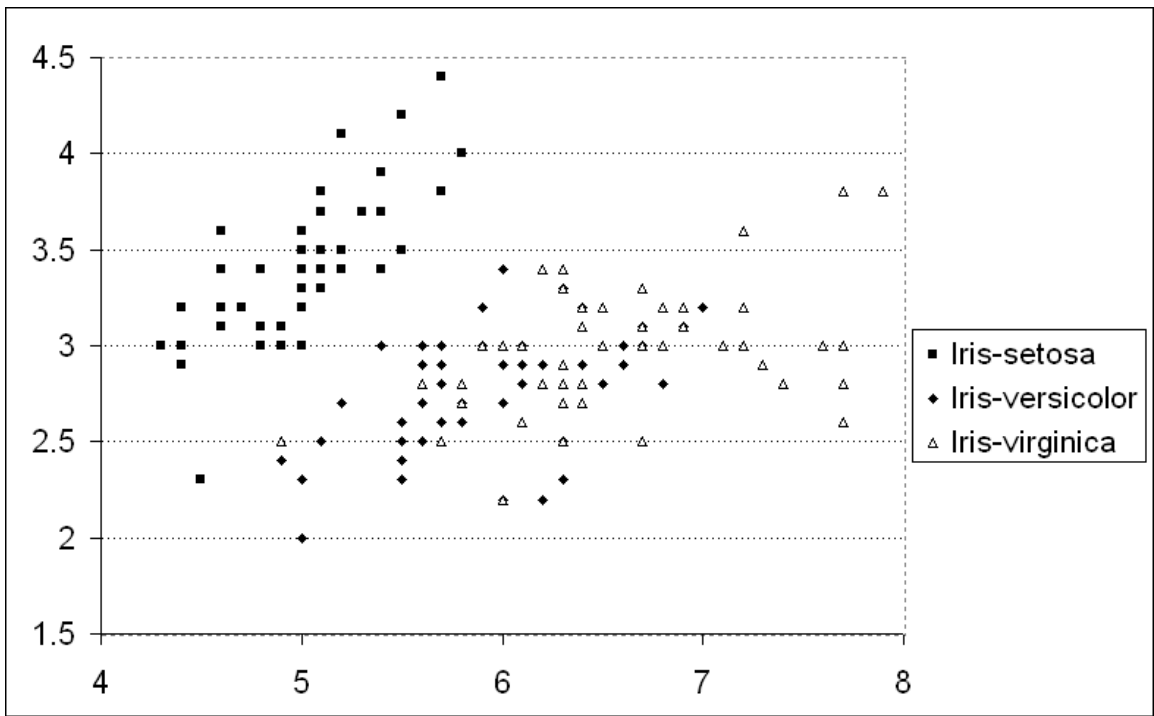
Pre ďalšiu prácu by bolo veľmi zaujímavé použiť podobný spôsob simulácie s iným modelom, s ktorým by sa dalo vyhnúť problémom spojeným s použitím JASTAPu. Taktiež by sa väčšia pozornosť mohla venovať reálnej aplikácii paralelizácie a preskúmaniu možností v tejto oblasti.

A Príloha: Iris

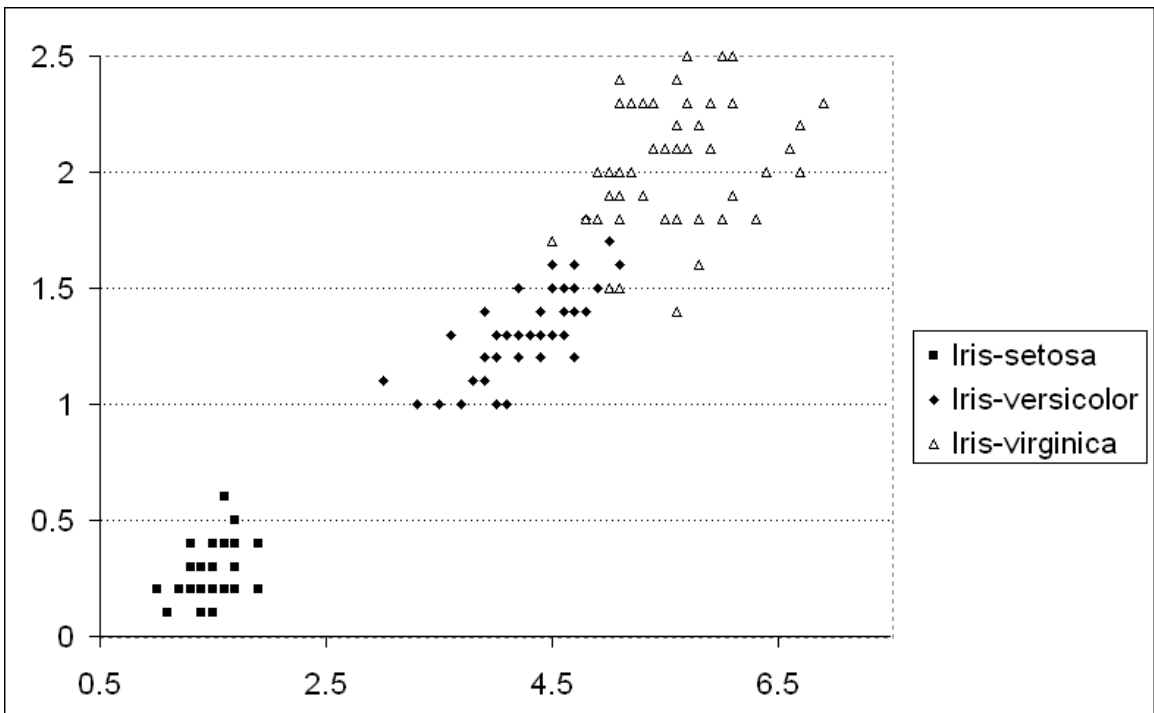
Na obrázkoch v prílohe vidieť rozloženie (distribúciu) jednotlivých kvetov z Iris databázy. Každý kvet je definovaný vektorom štvrtej dimenzie. Na prvom z dvojice obrázkov sú vykreslené prvé dve dimenzie a na druhom druhé dve.

Na (obr. A) je zobrazené rozloženie jednotlivých exemplárov Iris databázy **pred normalizáciou**. Ide o rozloženie vzhľadom na prvé dve dimenzie zo štyroch. Na (obr. B) je zobrazené rozloženie jednotlivých exemplárov Iris databázy vzhľadom na tretiu a štvrtú dimenziu **pred normalizáciou**.

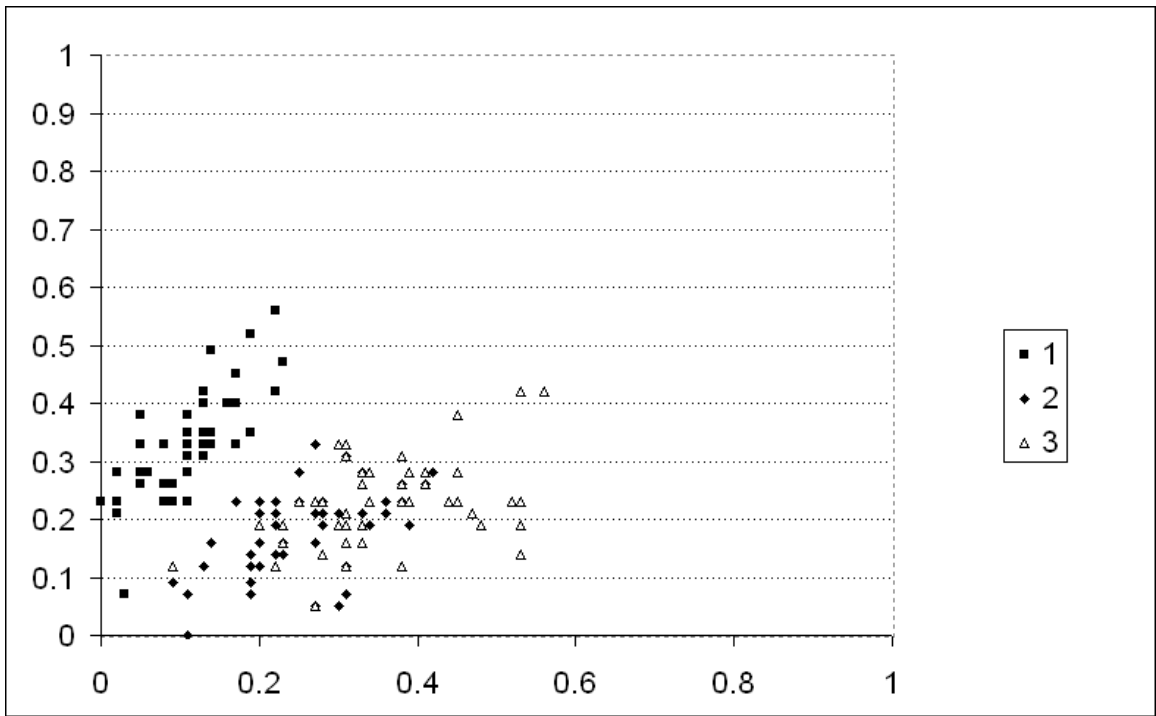
Na (obr. C) je zobrazené rozloženie jednotlivých exemplárov Iris databázy **po normalizácii** vzhľadom na prvé dve dimenzie zo štyroch. Na (obr. D) je zobrazené rozloženie jednotlivých exemplárov Iris databázy vzhľadom na tretiu a štvrtú dimenziu **po normalizácii**.



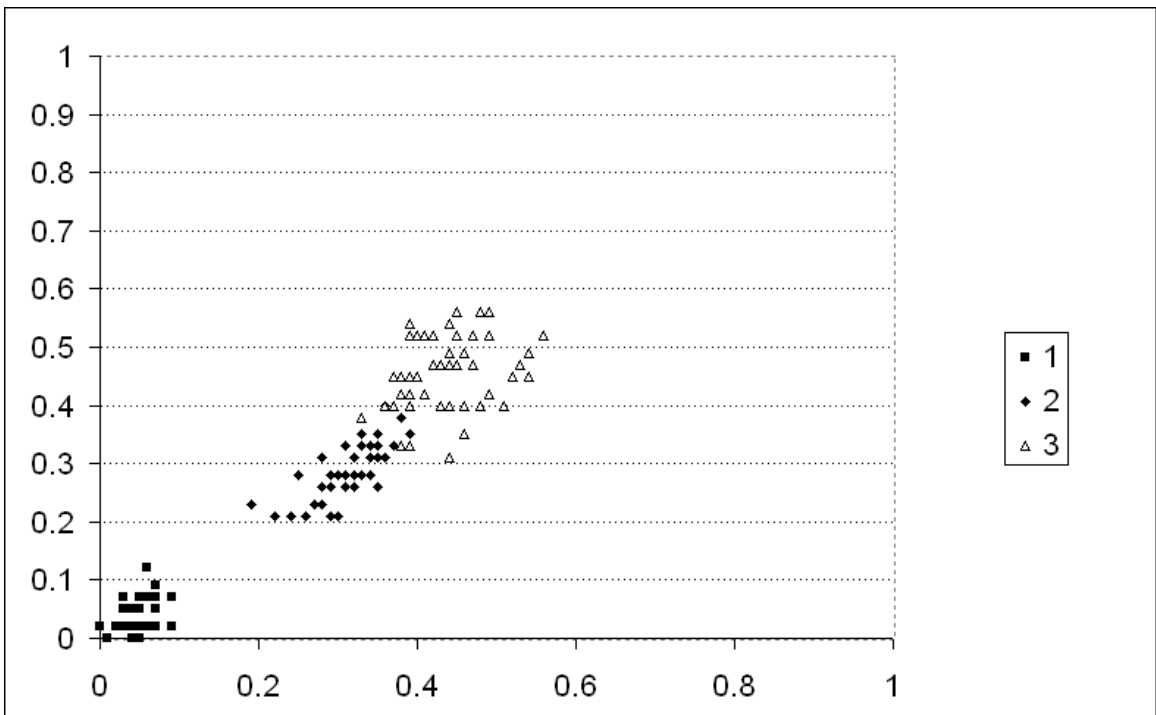
Obr. A



Obr. B



Obr. C



Obr. D

Referencie

- [1] W. GERSTNER, W. M. KISTLER: Spiking Neuron Models, (2002) Cambridge Univ. Press, Cambridge
- [2] E. M. IZHIKEVICH: Simple Model of Spiking Neurons, (2003) IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 14, NO. 6, NOVEMBER 2003
- [3] E. M. IZHIKEVICH: Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting., (2007) The MIT press
- [4] E. M. IZHIKEVICH: Polychronization: Computation with Spikes, Neural Computation 18, 245–282 (2006), ©2005 Massachusetts Institute of Technology
- [5] E. M. IZHIKEVICH, J. A. GALLY AND G. M. EDELMAN: Spike-timing Dynamics of Neuronal Groups, Cerebral Cortex V 14 N 8 ©Oxford University Press 2004;
- [6] J. JANČA, I. STAVROVKÝ, J. PAVLÁSEK: Modeling of neuronal functions: A neuron-like element with the graded response, Computers and Artificial Intelligence, vol. 13, pp. 603-620, 1994
- [7] J. PAVLÁSEK, J. JANČA, R. HARMAN: Rate coding: neurobiological network performing detection of the difference between mean spiking rates, 2003
- [8] M. VALKO, NUNO C. MARQUES, M. CASTELLANI: Evolutionary Feature Selection for Spiking Neural Network Pattern Classifiers, 2005
- [9] R. D. LAWRENCE, G. S. ALMASI, H. E. RUSHMEIER: A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Mining, Data Mining and Knowledge Discovery, Springer Netherlands 1999
- [10] B. RUF, M. SCHMITT: Self-Organization of Spiking Neurons Using Action Potential Timing, Institute for Theoretical Computer Science, Technische Universität Graz 1998
- [11] S. M. BOHTE, HAN LA POUTRÉ, J. N. KOK: Unsupervised Clustering With Spiking Neurons By Sparse Temporal Coding and Multilayer RBF Networks, IEEE Transactions on Neural Networks 2001

- [12] C. PANCHEV, STEFAN WERMTER: Hebbian Spike-Timing Dependent Self-Organization In Pulsed Neural Networks, Proceedings of World Congress on Neuroinformatics 2001
- [13] D. M. SALA, K. J. CIOS, J. T. WALL: Self-Organization in Networks of Spiking Neurons, University of Toledo and Medical College of Ohio
- [14] T. KOHONEN: Self-Organizing Maps, Third Extended Edition 2001
- [15] J. HAN, M. KAMBER: Data Mining – Concepts and Techniques, 2000 (c) Morgan Kaufmann Publishers
- [16] A. KARATZOGLOU, D. MEYER, KURT HORNIK: Support Vector Machines in R, Journal of Statistical Software, April 2006, Volume 15, Issue 9
- [17] I. FISCHER, J. POLAND: New Methods for Spectral Clustering, Hebrew University, Jerusalem, Israel, June 2004
- [18] P. PULKKINEN: Multiobjective identification of fuzzy models with hybrid methods, Master of Science Thesis, Tampere University of Technology, 2005
- [19] J. VREEKEN: Spiking neural networks, an introduction, Institute for Information and Computing Sciences, Utrecht University