UNIVERZITA KOMENSKÉHO V BRATISLAVE FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



MATEMATICKÉ MODELOVANIE SPRACOVÁVANIA INFORMÁCIÍ V MOZGU

DIPLOMOVÁ PRÁCA

Bc. Samuel ZBOJA

 $\mathbf{2020}$

UNIVERZITA KOMENSKÉHO V BRATISLAVE FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MATEMATICKÉ MODELOVANIE SPRACOVÁVANIA INFORMÁCIÍ V MOZGU

DIPLOMOVÁ PRÁCA

Študijný program:	Ekonomicko-finančná matematika a modelovanie
Študijný odbor:	1114 Aplikovaná matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	doc. RNDr. Peter Mederly, CSc.

Bratislava 2020

Bc. Samuel ZBOJA





Univerzita Komenského v Bratislave Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Samuel Zboja
Študijný program:	ekonomicko-finančná matematika a modelovanie
	(Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	aplikovaná matematika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický

Názov: Matematické modelovanie spracovávania informácií v mozgu Mathematical modelling of information processing in the brain

Anotácia: Matematické modelovanie predstavuje dôležitý nástroj pri snahe o charakterizovanie činnosti mozgu. Základnou úlohou mozgu je spracovávanie informácií prichádzajúcich z vonkajšieho prostredia, v ktorom sa človek nachádza, a vygenerovanie nervových podnetov pre adekvátne reakcie ľudského tela. Matematické modelovanie tohto spracovávania informácií umožňuje aspoň v istej obmedzenej miere nahliadnuť do nesmierne zložitej činnosti ľudského mozgu.

Vedúci:	doc. RNDr. Peter M	ederly, CSc.
Katedra:	FMFI.KAMŠ - Kate	dra aplikovanej matematiky a štatistiky
Vedúci katedry:	prof. RNDr. Daniel	Sevčovič, DrSc.
Dátum zadania:	08.01.2019	
Dátum schválenia:	08.01.2019	prof. RNDr. Daniel Ševčovič, DrSc. garant študijného programu

študent

vedúci práce

Poďakovanie

Týmto by som sa chcel poďakovať svojmu vedúcemu, doc. Petrovi Mederlymu, za veľkú ochotu, pomoc, spoluprácu a veľa odborných rád pri tvorbe tejto práce.

Abstrakt

Zboja, Samuel: Matematické modelovanie spracovávania informácií v mozgu [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; vedúci práce: Doc. RNDr. Peter Mederly, CSc., Bratislava, 2020, 122s.

V tejto diplomovej práci sa zaoberáme činnosťou mozgu, neurónmi a matematickými modelmi popisujúcimi ich činnosť. Hlavným cieľom práce je ilustrovať použitie matematického modelovania v oblasti neurovedy a vysvetliť základné princípy neurálneho kódovania, neurálneho dekódovania a neurónových sietí, a to aj pre čitateľa, ktorý nemá výraznejšie znalosti z oblasti neurovedy. Na dosiahnutie tohto cieľa najprv uvádzame základné pojmy z biológie a základné princípy fungovania mozgu a neurónov. Vysvetľujeme aj princíp spracovávania jednotlivých podnetov mozgom. Potom vysvetľujeme základné matematické pojmy popisujúce činnosť neurónov ako spike train alebo firing rate. Dalej predstavujeme základný kódovací model zvaný Linear-Nonlinear-Poisson model. Následne prezentujeme dekódovacie modely využívajúce najmä teóriu pravdepodobnosti a metódu maximálnej vierohodnosti. Spomínané modely aplikujeme na dostupné dáta, získané z experimentov, prípadne si dáta sami vygenerujeme pomocou simulačných metód a popíšeme ich výsledky. V poslednej časti práce sa venujeme prepojeniam medzi neurónmi a diferenciálnymi rovnicami, ktoré ich popisujú (tzv. firing rate modelom). Prezentujeme aj výsledky vlastných numerických simulácií a pomocou metód strojového učenia ukazujeme vplyv jednotlivých parametrov na riešenia spomínaných rovníc.

Kľúčové slová: Mozog, Neuróny, Neurálne kódovanie, Neurálne dekódovanie, Firing rate model

Abstract

Zboja, Samuel: Neurons and mathematics [Diploma Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: RNDr. Peter Mederly, CSc., Bratislava, 2020, 122p.

In this diploma thesis we discuss the activity of the brain, the neurons and the models describing them. The main goal of the thesis is to illustrate the use of mathematical models in the field of neuroscience and to explain basic concepts of neural encoding, neural decoding and neural networks even to a reader not disposing of significant knowledge in the field of neuroscience. For reaching this goal, we first sum up the basic terms from the biology as well as basic concepts of the behaviour of the brain and neurons. We also explain the principles of processing specific stimuli by the brain. Then we explain basic mathematical terms in neuroscience describing the activity of the neurons such as spike train or firing rate. Further, we introduce the basic encoding model called Linear-Nonlinear-Poisson model. Next we demonstrate decoding models mainly using probability theory and maximum likelihood method. We apply mentioned models to some available experimental data or, if required, we generate the data by ourselves using stochastic simulations and then describe their results. The last part of the thesis we dedicate to the connections between neurons and the differential equations describing them, also called as the firing rate model. We present the results of our own numerical simulations and, with the support of machine learning methods, we examine the effects of the parameters on the model.

Keywords: Brain, Neurons, Neural encoding, Neural decoding, Firing rate model

Obsah

Ú	vod			0
1	Uv	ede	nie do problematiky1	2
	1.1	Fu	nkcia mozgu1	2
	1.2	Ne	euróny1	2
	1.3	Pre	esun informácie z receptorov do mozgu1	4
	1.3.	1	Recepčné pole 1	5
	1.3.	2	Príklad zrakovej dráhy1	5
2	Ne	urál	lne kódovanie2	0
	2.1	Sti	imul2	0
	2.2	Od	lpoveď neurónu2	1
	2.2.	1	Spike train	1
	2.2.1	2	Firing rate a jej odhadovanie2	3
	2.2.	3	Pravdepodobnostné rozdelenie časov spikov2	7
	2.2.	4	Generovanie spike trainov2	9
	2.3	LN	VP model	0
	2.3.	1	Lineárny filter	0
	2.3.2	2	Odhad kernelovej funkcie	0
	2.3.5	3	Statická nelinearita	4
	2.3.4	4	Možné problémy LNP modelu	5
	2.3.	5	Zhrnutie LNP modelu	0
3	Ne	urál	lne dekódovanie4	2
	3.1	Vy	vsvetlenie pravdepodobností4	2
	3.2	Ex	periment s opicou4	3

	3.3	Dekódovací model45
	3.3.1	Optimálna rozhodovacia hranica46
	3.3.2	ROC krivka
	3.3.3	Generovanie ROC kriviek a numerické počítanie
	optir	nálnej hranice na základe rôznych hustôt50
	3.3.4	Strata za chybu
	3.4	Dekódovanie populácie neurónov54
4	Neu	rónové siete
	4.1	Synapsie
	4.2	Chemicky riadené kanály61
	4.2.1	Synaptická vodivosť62
	4.3	Krátkodobá synaptická plasticita64
	4.3.1	Synaptická depresia65
	4.3.2	Synaptická facilitácia65
	4.3.3	Dôsledky krátkodobej synaptickej plasticity66
	4.4	Firing rate model
	4.4.1	Synaptický vstup69
	4.4.2	Vstupno-výstupná funkcia71
	4.4.3	Rovnice popisujúce firing rate model73
	4.4.4	Rôzne interpretácie synaptického vstupu vo firing rate modeloch $\ldots .74$
	4.4.5	Synaptická saturácia a krátkodobá synaptická plasticita vo firing rate
	mode	eloch
	4.5 \$	šimulácie firing rate modelu pri jednej jednotke s rekurentným excitačným
	spojení	m

4.5.1 Model bez synaptickej saturácie, synaptickej plasticity a bez externého
vstupu
4.5.2 Model bez synaptickej saturácie, synaptickej plasticity a s externým
vstupom
4.5.3 Model so synaptickou saturáciou a bez synaptickej plasticity
4.5.4 Model so synaptickou saturáciou, synaptickou depresiou a bez
synaptickej facilitácie90
4.5.5 Model so synaptickou saturáciou a synaptickou facilitáciou 102
Zoznam použitej literatúry 109
Prílohy114

Úvod

Porozumenie ľudskej mysle sa stalo jednou z najväčších výziev pre vedu v 21. storočí. Snažíme sa porozumieť biologickej podstate vnímania, učenia sa, pamäti a vedomia. Ľudská myseľ je súbor najkomplexnejších procesov v celom vesmíre a pred sedemdesiatimi rokmi nebolo jej skúmanie do väčšej hĺbky ešte možné. Objavenie štruktúry DNA v roku 1953 znamenalo revolúciu v skúmaní tejto oblasti, pretože to viedlo k základnému pochopeniu ako informácie z génov riadia fungovanie buniek [11]. To umožnilo začať dôkladnejšie študovať ľudskú myseľ.

Ľudský mozog je zložitý orgán s vysokou výpočtovou kapacitou, ktorý konštruuje všetky naše myšlienky, pocity, spomienky. Doterajšie výskumy nasvedčujú tomu, že to, čo nazývame mysľou, je v podstate výsledkom funkcií vykonávaných mozgom.

Neuroveda (angl. *neuroscience*) sa zaoberá tým, čo nervový systém robí, ako funguje a prečo tak funguje. Neuroveda sa zaoberá nervovým systémom na rôznych úrovniach – od molekulárnej a bunkovej až po úroveň chovania sa a psychológie.

Vzhľadom na obrovskú komplexnosť mysle vedcom pri jej skúmaní nestačili len biologické a chemické experimenty, ale musela prísť na scénu aj matematika. Takto vznikla oblasť neurovedy nazvaná výpočtová neuroveda (angl. *computational neuroscience*). Výpočtová neuroveda za pomoci pokročilých matematických modelov skúma mnohé aspekty ľudskej mysle, pri ktorých bolo ešte donedávna nepredstaviteľné, že by sa dali popísať aj matematicky.

Jednou z vecí, ktorými sa výpočtová neuroveda zaoberá, je kódovanie a dekódovanie informácií, ktoré človek prijíma, mozgom. Ďalšiu vec, ktorú výpočtová neuroveda skúma, sú neurónové siete, čiže prepojenia medzi neurónmi. Skúmanie a vytvorenie matematického modelu týchto sietí viedlo k rozsiahlej revolúcií v umelej inteligencii (angl. *artifical inteligence*) a strojovom učení (angl. *machine learning*). Pri výpočtovej neurovede sa prepájajú poznatky z biológie

10

s poznatkami z matematiky, nakoľko bez teoretických znalostí štruktúry neurónov a bez matematických nástrojov by myseľ nebolo možné študovať a popisovať.

Aj napriek vyššie spomenutým pokrokom je ľudská myseľ stále veľmi tajomná a prebádaná len čiastočne. Práve preto je nutné oboznamovať vedeckú komunitu (ale aj laickú verejnosť) s doteraz dosiahnutými výsledkami, pretože porozumenie a výskum ľudskej mysle, vedie nielen k pokroku v počítačovej vede (vyššie spomenuté AI a ML), ale môže byť prínosom aj v medicíne.

V tejto práci sa budeme zaoberať kódovaním a dekódovaním informácií v mozgu a neurónovými sieťami.

Cieľom tejto práce bude uviesť čitateľa do základných pojmov v týchto oblastiach a podrobne opísať matematické modely, ktoré sa v nich využívajú. Výklad bude doplnený simuláciami modelov a postupov na vlastných dátach, ktoré si budeme v prípade potreby aj sami, vhodnými metódami, generovať. V každej kapitole budeme jednotlivé metódy, používané v modeloch, porovnávať, analyzovať ich citlivosť na zmeny parametrov a dopĺňať vlastnými postrehmi, komentármi a obrázkami.

V priebehu práce budeme vychádzať najmä z [5], [13], [22], [15].

V prvej kapitole stručne charakterizujeme činnosť mozgu a neurónov, vrátane spôsobu akým mozog prijíma a spracováva informácie.

V druhej kapitole vysvetlíme základné pojmy z neurovedy ako firing rate a spike train. Popíšeme aj základné modely kódovania.

V tretej kapitole predstavíme základné modely dekódovania.

V štvrtej kapitole vysvetlíme princíp neurónových sietí a modely, ktoré ich popisujú.

11

1 Uvedenie do problematiky

V tejto kapitole stručne charakterizujeme činnosť mozgu s dôrazom na spôsob, akým mozog prijíma informácie, ako ich spracováva a ako dáva ostatným častiam tela príkazy na reakciu. Budeme vychádzať najmä z [17], [14], [15], [26], [12], [3].

1.1 Funkcia mozgu

Všetky činnosti nášho tela, či už automatické ako dýchanie, tlkot srdca, trávenie a podobne, či ovládané vôľou, ako pohyb, rozprávanie a prijímanie potravy alebo to najkomplexnejšie – myslenie – sú riadené mozgom. Mozog riadi činnosti nášho tela najmä na základe vyhodnotenia podnetov prijatých tak z vonkajšieho, ako aj z vnútorného prostredia. Podnety sú zachytávané na receptoroch, ktoré ich transformujú na vstupnú informáciu do nervového systému. Receptory zachytávajúce vonkajšie podnety sa nachádzajú najmä v zmyslových orgánoch. Sú to napríklad tyčinky a čapíky na sietnici oka, chuťové bunky na jazyku alebo mechanické receptory vo vnútornom uchu. Podnety z vnútorného prostredia sú zachytávané receptormi distribuovanými v rôznych orgánoch tela, napríklad baroreceptory v stenách niektorých ciev snímajúce veľkosť krvného tlaku alebo proprioreceptory vo svaloch a šľachách snímajúce ich napätie.

Prijímanie podnetov aj vykonávanie činností sa preto väčšinou odohráva mimo mozgu. Ako sa teda prenesie zachytená informácia z receptorov do mozgu? Odpoveďou sú neuróny.

1.2 Neuróny

Neuróny sú základnými stavebnými bunkami nervovej sústavy. Od ostatných buniek sa odlišujú schopnosťou rýchlo šíriť signály na relatívne veľké vzdialenosti. V našom tele ich je približne 100 miliárd a sú navzájom poprepájané.

Na Obrázku 1.1 môžeme vidieť stavbu jedného neurónu.



Obrázok 1.1 – Stavba neurónu

Každý neurón prijíma, spravidla cez dendrity, chemické¹ impulzy od neurónov, na ktoré je napojený. Tieto impulzy spôsobujú zmeny membránového potenciálu, teda napätia na membráne neurónu. Ak sa toto napätie zvýši nad určitú hranicu, neurón vygeneruje tzv. akčný potenciál. Vygenerovanie akčného potenciálu znamená, že sa zmena napätia na membráne preniesla z dendritov na axón a pozdĺž neho až k jeho výbežkom (terminálom). Tieto výbežky sú napojené na dendrity ďalšieho neurónu cez spojenia zvané synapsie. Vygenerovanie akčného potenciálu spôsobí, že neurón cez synapsiu odovzdá chemický impulz ďalšiemu neurónu resp. neurónom a signál sa takto šíri ďalej. Podrobnejšie sa synapsiám budeme venovať v podkapitole 4.1 kapitoly 4.

Obrázok 1.1 sme zobrali z $\left[27\right]$ a následne preložili.

Jednotlivé akčné potenciály sú typovo pre všetky neuróny rovnaké a majú vždy prakticky rovnaký priebeh. V dôsledku toho môžeme pri akčnom potenciáli konkrétneho neurónu v konkrétnom čase posudzovať len binárny stav, či nastal alebo nenastal. Čo sa ale líšiť môže, je frekvencia s akou akčné potenciály (pre jeden konkrétny neurón) v určitom časovom úseku nastávajú. Práve táto frekvencia sa často pokladá za "intenzitu signálu", ktorý v danej chvíli neurón generuje. Čím vyššia je frekvencia, tým sa signál pokladá za intenzívnejší.

Akčné potenciály jednotlivých neurónov sú hlavným zdrojom informácií pre ľudský mozog a zároveň sú aj hlavným nástrojom na šírenie informácií v rámci mozgu.

 $^{^1}$ V menšom rozsahu sa vyskytujú aj iné spôsoby od
ovzdania impulzu ako chemické. Tými sa ale v práci zaoberať nebudeme.

V oblasti neurovedy, ktorou sa zaoberá táto práca, sa namiesto akčného potenciálu často používa výraz "spike". Tento výraz budeme v rovnakom význame ako spojenie "akčný potenciál" používať aj v našej práci².

1.3 Presun informácie z receptorov do mozgu

V predošlých podkapitolách sme uviedli, že mozog spracováva informácie, ktoré vzniknú na základe podnetov zachytených na receptoroch. Tak isto sme uviedli, že neuróny sú bunky, ktoré sú schopné navzájom si posielať nejaké signály. V tejto časti budeme podrobnejšie charakterizovať cestu informácie vygenerovanej receptorom na základe podnetu z vonkajšieho alebo vnútorného prostredia do mozgu.

Podľa [15] receptor zachytí fyzikálnu alebo chemickú zmenu v prostredí, napríklad zmenu intenzity zvuku, zmenu tlaku alebo chemickú zmenu vyvolanú prijímanou potravou na jazyku a transformuje ju na vstupnú informáciu do nervového systému, ktorá je štandardne vo forme zmeny elektrického potenciálu. Pokiaľ je receptor súčasťou neurónu, čo je obvyklý prípad, uvedená zmena elektrického potenciálu vedie k vygenerovaniu akčného potenciálu resp. série akčných potenciálov v tomto neuróne v závislosti od intenzity podnetu. Aj v prípade, že receptor nie je priamo súčasťou neurónu (príkladom sú niektoré mechanické receptory v koži, napríklad Merkelove bunky), je na nejaký neurón prepojený a jeho aktivita spôsobí vygenerovanie príslušných akčných potenciálov v tomto neuróne. V oboch prípadoch predstavuje daný neurón začiatok príslušnej nervovej dráhy, po ktorej sa informácia z receptora dostáva do mozgu.

Na ceste do mozgu, resp. do mozgovej kôry sa však informácia mení. Ako sme uviedli, na receptoroch sa zachytávajú len základné fyzikálne podnety alebo chemické podnety. V mozgu ale samozrejme pracujeme s oveľa komplexnejším

² Vzhľadom na štandardné používanie tohto anglického výrazu a tiež výrazov od neho odvodených ho nebudeme v tejto práci prekladať do slovenčiny, ale budeme s ním zaobchádzať ako so slovenským slovom a skloňovať ho podľa vzoru dub.

typom informácie. Napríklad, keď sa pozeráme na dom, tak ho vnímame ako dom a nie ako súbor rôzne zafarbených pixelov v zornom poli, alebo keď počujeme niekoho rozprávať, tak vnímame význam jednotlivých viet a nie len sekvenciu zvukov rôznej výšky. Táto komplexifikácia vzniká kombinovaním informácií od rôznych neurónov a následným kombinovaním novo vzniknutých kombinácií atď.

Zjednodušený proces takejto komplexifikácie si vysvetlíme v časti 1.3.2, najprv si ale zavedieme koncept recepčných polí (angl. *receptive fields*).

1.3.1 Recepčné pole

Recepčné pole neurónu je v súčasnej literatúre v oblasti neurovedy definované ako súhrn takých charakteristík podnetu, ktorý u daného neurónu spôsobí vygenerovanie akčného potenciálu, alebo nejakej (podprahovej) aktivity, ktorá zvýši, alebo zníži pravdepodobnosť vygenerovania akčného potenciálu. Recepčné pole neurónu je dané tým, na aké receptory alebo neuróny je neurón napojený. V pôvodnom, jednoduchšom chápaní, ako napr. v [18], ktoré sa však v niektorých zdrojoch ([15]) uvádza aj dnes, sa pod recepčným poľom neurónu rozumie plocha alebo priestor spojený so zmyslovým vnímaním (napríklad oblasť na povrchu tela alebo časť sietnice), v rámci ktorej aplikovanie špecifického podnetu (napríklad dotyku alebo svetla) vyvolá vygenerovanie akčného potenciálu v danom neuróne.

Ak je neurón napojený na receptory na sietnici, tak jeho recepčné pole môže byť napríklad svetlo v nejakom bode zorného poľa. Pre neuróny v mozgovej kôre bude recepčným poľom niečo oveľa komplexnejšie, napr. "les". Takéto recepčné pole by mohol mať neurón, ktorý je napojený na neuróny vizuálneho systému s recepčným poľom napr. "veľa zeleno-hnedých vysokých objektov v zornom poli" a neuróny čuchového systému s recepčným poľom "vôňa borovice".

1.3.2 Príklad zrakovej dráhy

V tejto časti uvedieme príklad, ktorý priblíži princíp skladania (komplexifikácie) informácie na ceste od receptorov do mozgovej kôry načrtnutý v predošlých podkapitolách. Zraková dráha začína na sietnici. Na sietnici máme neuróny typu "on-center, off-surround". Sú to neuróny, ktoré majú, ako napovedá ich názov, také recepčné pole, aké je znázornené na Obrázku 1.2. Tieto neuróny teda vygenerujú spike pokiaľ je v danej časti zorného poľa³ svetlo v strede tejto časti a tma okolo neho⁴. Na tieto neuróny sú napojené neuróny z časti zrakovej dráhy nazvanej "primary visual cortex" alebo V1⁵. Ich recepčné polia vyzerajú ako útvary na Obrázku 1.3, kde sú



Obrázok 1.2 - "On center, off surround" recepčné pole

znamienkom "+" vyznačené plochy zorného poľa, ktoré pri osvetlení stimulujú neurón a znamienkom "-" plochy zorného poľa, ktoré pri neosvetlení stimulujú neurón.

Na Obrázku 1.4 je ukázané, akým spôsobom možno útvary z Obrázku 1.3 vyskladať pomocou útvarov z Obrázku 1.2. Počas toho, ako sa informácia šíri ďalej, sa

 $^{^3}$ Každý neurón je v inej časti sietnice, a teda prislúcha aj inej časti zorného poľa.

⁴ Existuje aj typ "off-center, on-surround". Pre tento typ by sme ďalej postupovali analogicky.

⁵ Jednotlivým častiam nervovej sústavy sa venovať nebudeme. Stačí vedieť, že nervová sústava sa delí na centrálnu (CNS) a periférnu (PNS). Do PNS zaraďujeme neuróny na receptoroch a neuróny tvoriace nervy, ktoré tie na receptoroch spájajú s CNS. Do CNS zaraďujeme mozog a miechu. Mozog delíme na tzv. "nižšie" mozgové časti ako napríklad predĺžená miecha, thalamus alebo mozoček a "vyššie" mozgové časti ako je mozgová kôra. Pri mozgovej kôre budeme rozlišovať "vyššie" časti a "nižšie" časti. Na neurónoch s receptormi sú recepčné polia najprimitívnejšie, v mieche a nižších mozgových častiach zložitejšie a vo vyšších častiach mozgovej kôry najzložitejšie. Viac o stavbe mozgu v [29].



Obrázok 1.3 - Príklady recepčných polí vo V1



Obrázok 1.4 - Vyskladanie recepčného poľa vo V1 pomocou tých na sietnici

kombinovaním útvarov na Obrázku 1.3 vytvárajú ešte špecifickejšie tvary a z nich sa ďalším kombinovaním vytvárajú už obrazy konkrétnych predmetov.

Ako je vidieť na Obrázku 1.5, čím ďalej postupujeme po zrakovej dráhe (a teda čím sme bližšie k vyšším častiam mozgu), tým sú neuróny citlivejšie na komplexnejšiu informáciu. Neskôr sa už vizuálna časť vnímania mieša s ostatnými časťami a jednotlivým predmetom sa (aj na základe kontextu) priraďuje sémantický význam – vznikajú komplexnejšie vnemy – myšlienky.



Obrázok 1.5 - Vývoj vnímania obrazu od sietnice až po vyššie časti mozgovej kôry

Uvedený model sa podarilo veľmi preukazným spôsobom potvrdiť, keď sa pri operáciách mozgu⁶ experimentálne potvrdila citlivosť konkrétnych neurónov, v časti mozgu (parahipokampus) dôležitej pre zapamätanie si, na konkrétne obrázky známych ľudí. Na Obrázku 1.8 môžeme vidieť jedny z výsledkov takéhoto experimentu publikované v [16]. Citlivosť neurónu na daný obrázok je vyjadrená pomocou výšky stĺpcov na dolnom grafe pod každým obrázkom. Presnejší význam

 $^{^6}$ Niektoré fázy operácie moz
gu je možné absolvovať pri plnom vedomí pacienta, nakoľko na moz
gu necítime bolesť.



Obrázok 1.6 – Výsledky experimentu merajúceho citlivosť konkrétneho neurónu na rôzne obrázky.

grafov pod obrázkom vysvetľovať v tejto časti práce nebudeme, stačí vedieť, že čím vyššie sú stĺpce, tým je neurón na daný obrázok citlivejší. Vidíme, že neurón je citlivý výhradne na obrázky dvojice Brada Pitta a Jennifer Aniston spolu. Zaujímavé je aj to, že na obrázky Jennifer Aniston samotnej až tak citlivý nie je. Toto naznačuje veľmi úzku špecifikáciu daného neurónu.

Na Obrázku 1.7 sú zase ukázané výsledky rovnakého experimentu pre iný neurón. Tento neurón je zjavne citlivý na Pamelu Anderson. Zaujímavé je, že neurón je citlivý nie len na obraz Pamely Anderson, ale aj na jej napísané meno (obrázok 61). Toto tiež potvrdzuje vyššie spomínané kombinovanie rôznych vnemov a citlivosť neurónov vyšších častí mozgu na významovú stránku podnetu. Obrázky 1.6 a 1.7 boli prevzaté z [3].



Obrázok 1.7 – Výsledky experimentu merajúceho citlivosť konkrétneho daného na rôzne obrázky.

2 Neurálne kódovanie

V prvej kapitole sme predstavili koncept reakcie neurónov na nejaký podnet. Neurálne kódovanie je odvetvie neurovedy zaoberajúce sa tým, ako neuróny túto reakciu (vo forme spikov) na daný podnet kódujú. Budeme sa snažiť matematicky popísať vzťah medzi podnetom a reakciou a na základe toho zostrojiť model, pomocou ktorého budeme schopní predikovať reakciu daného neurónu pri nových podnetoch. Budeme vychádzať najmä z [5], [13], [10]

2.1 Stimul

"Podnet" spomínaný v predchádzajúcej kapitole sa v neurovede nazýva stimul (z angl. *stimulus*). Stimul môže byť prakticky úplne hocičo, čo sme schopní nejakým spôsobom pociťovať (z anglického pojmu pre slovo "pocit", *sensation*).

Pri každom stimule je z hľadiska modelovania nutné vybrať nejakú jeho vlastnosť (alebo vlastnosti), ktorú vieme dobre kvantifikovať a merať. Túto vlastnosť budeme nazývať parameter stimulu. Slovným spojením "hodnota stimulu" budeme nazývať práve hodnotu jeho vybraného parametra.

V neurovede sa pri pokusoch najčastejšie používa ako stimul nejaký základný fyzikálny jav, napríklad svetlo alebo zvuk. Dôvodov, prečo sa volia práve takéto jednoduché stimuly, je niekoľko:

 Pre jednoduché stimuly je všeobecne známe, ktoré neuróny na ne reagujú. Napríklad v prípade svetla sú to (v časti 1.3.2) spomínané neuróny na sietnici. Vďaka tomu vieme, reakciu ktorých neurónov máme sledovať. Pri zložitejších stimuloch je veľmi náročné a často až nemožné identifikovať neuróny, ktoré ich kódujú. Je to spôsobené tým, že takéto neuróny sa nachádzajú vo vyšších častiach mozgu, kde sa prepája veľmi veľa rozličných vnemov, a teda sú veľmi špecifické.
 Fyzikálne javy vieme dobre parametrizovať (kvantifikovať a merať). Ak by bol neurón citlivý napríklad na orientáciu (naklonenie) vertikálnej svetelnej tyče v tmavom priestore, vieme zvoliť ako parameter stimulu uhol naklonenia tyče. Zatiaľ čo, ak by bol neurón citlivý špecificky na to, keď sa pozeráme na svoj dom, bolo by veľmi zložité takýto stimul parametrizovať.

Vzhľadom na tieto fakty sa budeme v najbližších kapitolách zaoberať výhradne reakciami na takýto jednoduchý typ stimulu.

Stimul budeme neurónu "púšťať" na časovom úseku dĺžky T a jeho hodnotu v čase t budeme označovať ako s(t). Táto hodnota môže byť konštantná alebo sa môže na časovom úseku meniť.

Časový úsek *T* uvažujeme v teórii ako spojitý. V počítačových výpočtoch však budeme počítať s jeho diskrétnymi hodnotami pri veľmi jemnom delení.

2.2 Odpoveď neurónu

Reakcia neurónu na podnet spomínaná v predchádzajúcej kapitole sa v neurovede nazýva odpoveď (z angl. *response*). Ako už bolo naznačené, neuróny odpovedajú na stimuly generovaním spikov. Čím vyššia je intenzita stimulu a čím viac stimul korešponduje s recepčným poľom neurónu, tým intenzívnejšie neurón spiky generuje.

Pre zjednodušenie sa budeme pozerať na odpoveď len jedného neurónu. Jeho odpoveď budeme popisovať veličinami "spike train" a "firing rate", o ktorých si povieme viac v nasledujúcich častiach.

2.2.1 Spike train

Podľa ([5],[22]) sú spiky reprezentovateľné ako diskrétne body na časovej osi. Pre daný časový úsek, stimul a neurón vieme zapísať časy spikov ako množinu



Obrázok 2.1 - Grafické znázornenie spike trainu

 $\{t_1, t_2, \dots, t_n\}$, kde *n* je počet spikov a t_i je čas *i*-teho spiku pričom $t_i \leq T$. Takáto množina sa nazýva "spike train"⁷.

Spike train sa zvykne graficky znázorňovať tak ako na Obrázku 2.1.

Matematicky vieme jeden konkrétny spike train vyjadriť ako ([5]):

$$arrho(t) = \sum_{i=1}^n \delta(t-t_i)$$
, (2.1)

kde $\delta(\mathbf{x})$ je Diracova delta funkcia ([28]). Funkcia $\varrho(t)$ sa nazýva funkcia neurálnej odpovede (angl. *neural response function*).

Presné časy, kedy nastanú spiky, nevieme deterministicky určiť. Je to spôsobené jednak tým, že pri experimentálnych podmienkach nevieme neurón nikde úplne izolovať od okolitých vplyvov, a tak isto tým, že každý neurón má nejaký vnútorný biofyzikálny stav, ktorý sa v priebehu času mení. Rozloženie jednotlivých časov spiku v rámci spike trainu teda musíme modelovať stochasticky.

Casy spikov sú teda náhodné diskrétne body na spojitom časovom úseku, a teda pravdepodobnosť vygenerovania spiku je v každom jednotlivom časovom bode nulová⁸. Tak isto je nulová pravdepodobnosť, že pri opakovaní pokusu (hoci aj pri úplne rovnakých podmienkach) dostaneme rovnaký spike train. Odpoveď vo forme spike trainu preto nie je jednoznačná. Takéto náhodné rozmiestnenie udalostí (bodov na časovej osi) sa nazýva *bodový proces*. Definíciu a vlastnosti bodového procesu môžeme nájsť v [30]. Navyše, pokiaľ sú jednotlivé udalosti bodového procesu od seba nezávislé, dostávame *Poissonov proces* [31]. V zvyšku práce budeme predpokladať, že časy spikov sú Poissonovým procesom⁹.

⁷ Slovenský preklad slova spike train "spike séria" nie je veľmi rozšírený, a preto budeme v práci používať anglickú verziu tohto pojmu.

⁸ Je dôležité poznamenať, že fakt, že jav má nulovú pravdepodobnosť neznamená, že nemôže nastať. V tomto prípade taký jav nastať môže, ale tým, že množina udalostí je nespočítateľná, každý konkrétny jav má nulovú pravdepodobnosť.

⁹ V realite môže vygenerovanie spiku ovplyvniť správanie neurónu a prispieť k vygenerovaniu ďalších alebo naopak, utlmiť neurón a zastaviť ďalšie generovanie. My ale budeme predpokladať nezávislosť všetkých spikov.

Pri Poissonovom procese je rozmiestnenie bodov síce náhodné, ale riadi sa Poissonovým pravdepodobnostným rozdelením. Na popísanie tohto rozdelenia nám slúži tzv. *firing rate*¹⁰.

2.2.2 Firing rate a jej odhadovanie

V neurovede (často aj v rámci jedného zdroja) sa pojmom "firing rate" označuje viacero rôznych veličín. V našej práci budeme pod týmto pojmom označovať veličinu z nasledovnej definície:

Definícia 1. Funkciu $r(t) = \lim_{\Delta t \to 0} \frac{p(t)}{\Delta t}$, kde p(t) je pravdepodobnosť nastania spiku na intervale $(t - \Delta t/2, t + \Delta t/2)$ pri stimule s(t), nazývame firing rate.

Takáto definícia firing rate je implicitne uvedená v [5]. Firing rate z Definície 1 budeme v práci niekedy nazývať aj "skutočná firing rate".

V prípade, že daný stimul nespĺňa žiadnu z charakteristík definujúcich recepčné pole skúmaného neurónu, budeme predpokladať, že príslušná $r(t) = 0^{11}$.

Ak je prítomný konštantný stimul (s(t) = s), ktorý súvisí s recepčným poľom skúmaného neurónu, budeme v práci predpokladať, že firing rate r(t) bude tiež konštantná. Čiže bude platiť r(t) = r.

Vtedy (ale nie len vtedy) má namiesto nami definovanej firing rate zmysel počítať počet spikov za jednotku času (v danom pokuse) (angl. *spike-count rate*):

$$\dot{r} = \frac{n}{T} = \frac{1}{T} \int_0^T \varrho(\tau) \ d\tau.$$
(2.2)

Počet spikov za jednotku času sa viaže na jeden beh pokusu. Vzhľadom na náhodnosť jednotlivých spike trainov v rámci pokusov popísanú v časti 2.2.1 nám lepší odhad firing rate r dáva tzv. priemerná firing rate¹² (angl. *average firing rate*):

 $^{^{10}}$ Rovnako ako pri spike train, preklad slova firing rate "intenzita generovania" nie je veľmi rozšírený, a preto budeme v práci používať anglickú verziu tohto slova v ženskom rode, bez skloňovania.

¹¹ Toto v skutočnosti platiť nemusí, neuróny zvyknú generovať spiky aj pokiaľ nie je prítomný žiadny stimul. V tejto kapitole sa takto generovaným spikom venovať nebudeme.

 $^{^{12}}$ Slovné spojenie "priemerná firing rate" reprezentuje jeden samostatný pojem. Nemyslí sa pod tým priemer z firing rate z Definície 1.

$$\hat{r} = \langle r \rangle = \frac{\langle n \rangle}{T} = \frac{1}{T} \int_0^T \langle \varrho(\tau) \rangle d\tau.$$
 (2.3)

Kde

$$\langle \varrho(\tau) \rangle = \frac{1}{k} \sum_{i=1}^{k} \varrho_i(\tau) \,. \tag{2.4}$$

Operátor $\langle \rangle$ u nás značí, že všetky funkcie neurálnej odpovede $\varrho_i(\tau)$ (čiže spike trainy) boli vygenerované pri rovnakom stimule.

Je dôležité poznamenať, že vzťah (2.3) vzniká kombináciou vzťahov (2.4) a (2.2). Priemeruje sa teda 2 krát. Najprv sa spriemerujú spike trainy z k pokusov (pri rovnakom stimule) , tak dostaneme $\langle \varrho(\tau) \rangle$ (vzťah (2.4)). Potom sa z tohto spriemerovaného spike trainu počíta počet spikov za jednotku času podľa (2.3), čím dostávame priemernú firing rate. Vzhľadom na vyjadrenie spike trainu pomocou (2.1) a ich spriemerovaniu pomocou (2.4) je zrejmé, že $\langle n \rangle$ bude vo vzťahu (2.3) predstavovať priemerný počet spikov na pokus.

Pre konštantnú firing rate r vieme následne vyjadriť vzťah medzi hodnotou (konštantného) stimulu s a priemernou firing rate $\langle r \rangle$ tak, že pre každú hodnotu stimulu s pustíme stimul s touto hodnotou dostatočne (v teórii nekonečne) veľakrát a pre každú túto hodnotu vypočítame príslušnú $\langle r \rangle$ pomocou vzťahu (2.3)¹³. Funkcia reprezentujúca tento vzťah sa nazýva *tunning curve* a vieme pomocou nej zistiť, na ktoré hodnoty daného stimulu je daný neurón najcitlivejší. Tunning curve budeme značiť ako $\varphi(s)$ a jej hodnoty budú vyjadrené v hertzoch (spiky za sekundu).

Častejšie sa však stretávame s neurónmi, ktoré nie sú citlivé na konkrétne hodnoty stimulu, ale na ich zmeny v čase. Vtedy je na generovanie spikov nutné neurónu púšťať nekonštantný stimul s(t).

V takom prípade je aj firing rate r(t) v čase nekonštantná. Jej odhadovanie pomocou priemernej firing rate $\langle r \rangle$ teda nemá veľký zmysel, pretože sa pri ňom stráca

¹³ Vo vzťahu na výpočet $\langle r \rangle$ sa *s* nikde nevyskytuje, treba si ale uvedomiť, že $\langle \varrho(\tau) \rangle$ sa viaže na jeden konkrétny stimul *s* (dokonca na jednu jeho konkrétnu hodnotu, keďže v tomto prípade je stimul konštantný), čiže $\langle r \rangle$ v konečnom dôsledku od *s* závisí. Zápis $\langle r(s) \rangle$ sa však v literatúre nepoužíva.

informácia o tom, pri akom správaní sa stimulu nastávali spiky častejšie (čiže r(t) mala vyššie hodnoty) a pri akom menej často (čiže r(t) mala nižšie hodnoty).

Spôsobov ako odhadnúť v čase nekonštantnú firing rate je viacero (pozri [5] str. 12-14 alebo [13] obr. 3.2). Jednou z najčastejšie používaných metód na odhadnutie firing rate je metóda, pri ktorej pre každý čas t spočítame priemerný počet spikov na časovom okne dĺžky wt okolo tohto času. Opäť musíme neurónu pri každom pokuse púšťať úplne rovnaký (tento raz už ale nie konštantný) stimul. Odhad firing rate bude teda vyzerať nasledovne:

$$\widehat{r}(t) = \frac{1}{wt} \int_{t-wt/2}^{t+wt/2} \langle \varrho(\tau) \rangle \ d\tau.$$
(2.5)

Pre nekonečne veľa pokusov a $wt \rightarrow 0$ sa $\hat{r}(t)$, odhadnuté vzťahom (2.5), blíži k skutočnému r(t) z Definície 1.

V praxi je však počet pokusov obmedzený, a teda je nutné pri odhadovaní r(t) pomocou vzťahu (2.5) vybrať vhodnú veľkosť *wt*. Ak by bolo *wt* príliš malé, pre veľa hodnôt *t* by bola hodnota $\hat{r}(t)$ nulová a stratili by sme schopnosť rozlíšiť výšku hodnôt r(t). Odhad firing rate by sa tak začínal podobať skôr na $\langle \varrho(t) \rangle$. Naopak, ak by bolo *wt* príliš veľké, interval (t - wt/2, t + wt/2), na ktorom by sme počítali $\hat{r}(t)$ by bol príliš široký a odhad firing rate by nezachytával prudšie zmeny vo firing rate na krátkych časových úsekoch. Odhad firing rate by sa tak začínal podobať skôr na $\langle r \rangle$. Dalo by sa povedať, že zväčšovanie *wt* zhladzuje $\hat{r}(t)$.

Úvahy o $\hat{r}(t)$ a wt z predošlého odseku si teraz otestujeme. Na Obrázkoch 2.2, 2.3 sú výsledky našich vlastných simulácií neurónu generujúceho spiky. Pri simuláciách sme uvažovali neurón s firing rate $r(t) = 4 \sin (4t^2) + 4$ a interval dĺžky T = 2s.

Pri zvolenej firing rate sme najprv vygenerovali k = 200 spike trainov (to budú naše dáta)¹⁴. Z nich sme (pre ilustráciu) na základe (2.4) resp. (2.3) vypočítali priemernú firing rate $\langle r \rangle$. V hornom grafe Obrázku 2.2 je zakreslená r(t) a $\langle r \rangle$.

 $^{^{14}}$ Tento krok je čisto technický a vytvára dáta, ktoré sa v realite získavajú pokusmi, pretože skutočná r(t)nikdy známa nie je. Postup generovania spike trainov na základe zvolenej firing rate je vysvetlený v podkapitole 2.2.4

V dolnom grafe Obrázku 2.2 je zakreslených náhodne vybraných 30 spike trainov spomedzi tých 200 vygenerovaných, pričom každá úroveň y-ovej osi zodpovedá jednému spike trainu. Takýto graf viacerých spike trainov sa nazýva *raster plot*. Môžeme vidieť, že tam kde sú hodnoty r(t) vyššie, sa aj generuje viac spikov. Tento fakt vyplýva priamo z definície r(t).

Na Obrázku 2.3 sú odhady firing rate z vygenerovaných spike trainov, vykonané na základe (2.5), porovnané so skutočnou hodnotou r(t). Veľkosť okna wtsme volili postupne ako 2s, 110ms, 16ms a 2ms. Ukazuje sa presne to, čo sme predpokladali. Zvyšovanie wt vyhladzuje $\hat{r}(t)$ a pri vysokých hodnotách (horný graf) sa stráca informácia o zmenách v r(t) a graf $\hat{r}(t)$ sa začína podobať na $\langle r \rangle$. Naopak, pri nízkych hodnotách wt (dolný graf) je interval, na postupne ktorom sa počíta $\hat{r}(t)$ veľmi úzky, a preto pravdepodobnosť výskytu spiku nízka. Preto má $\hat{r}(t)$ často



Obrázok 2.2 - Vykreslená r
(t) a na jej základe vypočítaná $\langle r\rangle$ a vygenerované spike trainy

nulové hodnoty.



Na záver tejto časti poznamenávame, že firing rate r(t) nie je v žiadnom prípade pravdepodobnosť nastania spikov v jednotlivých bodoch (tá je pre každý bod nulová). Tak isto to ani nie je hustota pravdepodobnosti. Treba ju chápať iba v zmysle Definície 1 a pre intuitívne pochopenie stačí vziať na vedomie, že čím vyšší je integrál r(t) na určitom intervale, tým vyšší je očakávaný počet spikov na danom intervale.

2.2.3 Pravdepodobnostné rozdelenie časov spikov

V časti 2.2.2 sme ukázali, ako sa dá pomocou jednotlivých spike trainov odhadnúť firing rate. V tejto časti si ukážeme, aké pravdepodobnostné rozdelenie majú jednotlivé časy spikov v rámci spike trainu v závislosti od firing rate.

Pri generovaní spike trainov budeme predpokladať, že firing rate r(t)poznáme. V časti 2.2.1 sme si povedali, že spike train je Poissonov proces, ktorý generuje náhodné body na časovej osi v závislosti od hodnôt r(t). Ak je r(t)konštantná (r(t) = r) (čiže máme aj konštantný stimul), tak sa príslušný Poissonov proces nazýva homogénny.

Teraz si odvodíme pravdepodobnosť nastania n spikov pri konštantnej firing rate r (čiže pre homogénny Poissonov proces) na časovom úseku dĺžky T. Najprv si tento úsek rozdelíme na M častí, každá s dĺžkou Δt . Predpokladáme, že pre dostatočne malé Δt môže na jednom úseku nastať maximálne 1 spike. Podľa Definície 1 je preto pre $\Delta t \rightarrow 0$ na každom úseku pravdepodobnosť spiku rovná $r\Delta t$. Nastanie spiku je teda na intervaloch binomický jav a bude platiť:

$$P(n \text{ spikov}) = \lim_{\Delta t \to 0} {\binom{M}{n}} (r\Delta t)^n (1 - r\Delta t)^{(M-n)}.$$
(2.6)

Keďže $M = T/\Delta t$ tak pre malé $\Delta t \rightarrow 0$ je M neohraničené a môžeme položiť $M - n \approx M$ a $M!/(M - n)! \approx M^n$. Potom s využitím $\lim_{x \to 0} (1 + x)^{\frac{1}{x}} = e$ môžeme vzťah (2.6) prepísať na:

$$P(n \text{ spikov}) = \frac{(rT)^n}{n!} e^{-rT}.$$
(2.7)

Teraz definujme $P(\{t_1, t_2, ..., t_n\})$ ako pravdepodobnosť, že vygenerujeme n spikov a zároveň, že sa v každom z intervalov $(t_1 - \delta t/2, t_1 + \delta t/2), (t_2 - \delta t/2, t_2 + \delta t/2), ..., (t_n - \delta t/2, t_n + \delta t/2)$ bude nachádzať práve jeden z nich. Potom je zrejmé, že

$$P(\{t_1, t_2, ..., t_n\}) = P(n \text{ spikov}) \int_{t_n - \delta t/2}^{t_n + \delta t/2} \dots \int_{t_2 - \delta t/2}^{t_2 + \delta t/2} \int_{t_1 - \delta t/2}^{t_1 + \delta t/2} g(t_1, t_2, ..., t_n) dt_1 dt_2 \dots dt_n \quad (2.8)$$

pričom g je združená hustota rozdelení jednotlivých časov spikov na intervale (0, T)ⁿ. Keďže máme homogénny Poissonov proces, tak každá kombinácia časov spikov { $t_1, t_2, ..., t_n$ } má rovnakú pravdepodobnosť, a preto hustota g musí mať v každom bode intervalu (0, T)ⁿ rovnakú hodnotu (a inde nulovú). Naviac musí platiť, že

$$\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(t_1, t_2, \dots, t_n) dt_1 dt_2 \dots dt_n = 1.$$
 (2.9)

Z predošlých faktov vyplýva, že $g(t_1, t_2, ..., t_n) = 1/T^n$. Vzhľadom na tento fakt spojením (2.7) a (2.8) dostávame:

$$P(\{t_1, t_2, ..., t_n\}) = \frac{(rT)^n}{n!} e^{-rT} \left(\frac{\delta t}{T}\right)^n.$$
(2.10)

Ešte treba poznamenať, že napr. $\{t_1=1, t_2=2, t_3=3\}$ a $\{t_1=2, t_2=1, t_3=3\}$ je ten istý spike train. Preto $P(\{t_1, t_2, ..., t_n\}) = P(\{t_1, t_2, ..., t_n\}) + ... + P(\{t_n, t_{n-1}, ..., t_1\})$, kde "…" medzi "+" reprezentuje P pre každú permutáciu $t_1, t_2, ..., t_n$. Preto treba výraz na pravej strane rovnice (2.10) prenásobiť ešte n!.

2.2.4 Generovanie spike trainov

Na záver podkapitoly 2.2 si ukážeme jednoduchý algoritmus, ktorým vieme generovať spike trainy neurónu s firing rate r(t).

Budeme vychádzať z Definície 1. Najprv si zvolíme dĺžku časového kroku Δt . V Definícii 1 predpokladáme, že $\Delta t \rightarrow 0$, preto pokiaľ chceme dostať vierohodné výsledky, musíme zvoliť Δt dostatočne malé. My budeme voliť $\Delta t = 0.005$ ms. Vytvoríme si časový vektor samých núl, zodpovedajúci intervalu (0, *T*), ktorý bude mať dĺžku $M = T/\Delta t$.

Každá zložka časového vektora bude prislúchať určitej časovej hodnote t_i a bude nadobúdať hodnotu 1 alebo 0 podľa toho, či na intervale (t_i , $t_i + \Delta t$) nastal, alebo nenastal spike.

V cykle budeme prechádzať jednotlivé zložky časového vektora a porovnávať $r(t_i)\Delta t$ s U_i , pričom U_i bude realizácia náhodnej premennej $U \sim R(0,1)$. Ak $r(t_i)\Delta t$ > U_i , tak prepíšeme *i*-tu zložku časového vektora na 1.

Výsledný binárny časový vektor bude reprezentovať 1 realizáciu spike trainu neurónu s firing rate r(t) spĺňajúcou Definíciu 1.

Pri vytváraní tohto algoritmu sme boli inšpirovaní cvičením v [13].

V časti 2.2.2 sme si ukázali, ako vieme pre jeden konkrétny stimul na základe spike trainov odhadnúť firing rate neurónu. Čo ak by sme ale chceli pre nejaký stimul vedieť odhadnúť firing rate neurónu ešte predtým, ako neurón vygeneruje spiky?

V nasledovnej časti sa pokúsime zostrojiť všeobecný model, ktorý ukazuje závislosť stimulu a odpovede. V neurovede sa pre tento účel podľa [5] využíva tzv. *Linear nonlinear Poisson* model (LNP model) použitý aj v [2].

Pod odpoveďou neurónu na daný stimul budeme (pokiaľ nebude uvedené inak) od teraz myslieť jeho firing rate na danom časovom úseku.

2.3.1 Lineárny filter

Najprv budeme vychádzať z predpokladu, že hodnotu odpovede v čase t vieme vyjadriť ako lineárnu kombináciu hodnôt stimulu v časoch $t_1, t_2, ..., t_n$ kde $t_i < t$. Pre spojitý čas teda môžeme odhadnúť firing rate pomocou vzťahu

$$\widehat{r}(t) = \int_0^{\tau_{max}} s(t-\tau) D(\tau) d\tau, \qquad (2.11)$$

kde $\tau_{\text{max}} > 0$ a $D(\tau)$ je tzv. kernelová funkcia, ktorá nám hovorí o tom, ako silno a s akým znamienkom prispieva hodnota stimulu v danom čase pred časom tk odpovedi v čase t. Integrál vo vzťahu (2.11) sa nazýva lineárny filter.

Pre zjednodušenie budeme predpokladať iba stimuly, ktoré sa "pohybujú okolo nuly", čiže platí $\int_0^T s(t) dt \approx 0$. Takýto predpoklad je v poriadku z hľadiska nášho modelu, pretože (ako sme spomenuli v časti 2.2.2) neuróny sú zvyčajne citlivé najmä na zmeny v stimule a nie na jeho absolútne hodnoty. Tak isto z hľadiska voľby stimulu je tento predpoklad v poriadku, pretože aj pre parameter stimulu, ktorý nemôže byť záporný, vieme vždy odpočítať jeho priemernú hodnotu, a tým zaručiť (približnú) nulovosť integrálu stimulu.

2.3.2 Odhad kernelovej funkcie

Ako bolo povedané v kapitole 1, neuróny sú citlivé na rôzne stimuly, pričom pod rôznymi stimulmi sa nemyslí len typ stimulu (svetlo, zvuk apod.), ale aj priebeh (správanie sa) daného stimulu v čase. Niektoré neuróny môžu byť napr. citlivé na prudký nárast stimulu, iné zase na prudký nárast nasledovaný miernym poklesom a podobne. V nasledujúcich častiach sa nebudeme zaoberať ani typom stimulu, ani ako je kvantifikovaný, ale bude nás zaujímať iba priebeh jeho hodnôt v čase (bez ohľadu na to, čo tieto hodnoty reprezentujú).

Zo vzťahu (2.11) vyplýva, že čím podobnejšie sú funkcie D a s na určitom danom úseku pred časom t, tým vyššiu hodnotu bude odhad \hat{r} v čase t nadobúdať. Preto je intuitívnou voľbou funkcie D taká funkcia, ktorá vyzerá rovnako ako stimul, na ktorý je neurón najcitlivejší.

To, že je neurón na určitý stimul citlivý (definotoricky) znamená, že má pri ňom vyššiu firing rate, čiže aj vyššiu pravdepodobnosť vygenerovania spiku. Môžeme teda predpokladať, že jednotlivým spikom bude najčastejšie predchádzať taký priebeh stimulu, na ktorý je neurón citlivý.

Vzhľadom na tieto tvrdenia sa v neurovede "stimul, na ktorý je neurón najcitlivejší" (a teda aj D) odhaduje ako tzv. *spike-triggered average* (STA) :

$$D(\tau) = \operatorname{STA}(\tau) = \frac{1}{n} \sum_{i=1}^{n} s(t_i - \tau)$$
 (2.12)

pre $au_{\max} > au > 0.$

Výpočet spike-triggered averageu prebieha nasledovne:

1. Neurónu sa pustí veľmi dlhý, náhodný
 15 stimul a zaznamenajú sa časy spikov
 $t_1,\ t_2,\ ...,\ t_n$

2. Pre každý čas spiku t_i uvažujeme časť stimulu, ktorá mu predchádzala, dĺžky τ_{max} . Pri voľbe τ_{max} predpokladáme, že hodnoty, ktoré nadobúdal stimul pred jednotlivými spikmi v minulosti vzdialenejšej ako τ_{max} sekúnd, už na ich nastanie nemajú žiaden vplyv. Môže sa zvoliť aj τ_{max} neobmedzené.

3. Následne všetky časti stimulu z predošlého bodu spriemerujeme.

Grafický príklad výpočtu spike-triggered average je na Obrázku 2.4, ktorý sme prevzali z [3].

 $^{^{15}}$ "Náhodnosť" takéhoto stimulu vysvetlíme neskôr.



Obrázok 2.4 – Ilustračné znázornenie priebehu výpočtu STA.

Dlhý náhodný stimul, ktorý sa púšťa neurónu, a na základe ktorého sa počíta spike-triggered average, sa zvykne voliť ako biely šum¹⁶. Biely šum je podľa [32] stochastický proces, ktorého každá hodnota v čase je generovaná nezávisle od ostatných z rozdelenia N(0,1). Takýto výber stimulu nám zaručuje, že pri dostatočnej dĺžke stimulu budú všetky spôsoby priebehu stimulu zastúpené v približne rovnakom pomere. Takže výsledný STA vierohodne reprezentuje priebeh stimulu, na ktorý je neurón citlivý.

Pri počítačových simuláciách simulujeme takýto stimul ako schodovitú funkciu, pričom každý schod má dĺžku Δt (jemnosť delenia osi x). Takýto stimul je znázornený na Obrázku 2.4.

V [5] je dokázané, že pri danom stimule *s* je *r* odhadnuté pomocou (2.11) s voľbou kernelovej funkcie ako spike-triggered average optimálne. Pod optimálnym sa myslí, že sa pri jeho voľbe minimalizuje chyba

$$E = rac{1}{T} \int\limits_{0}^{T} \left(\widehat{r}(t) - r(t)
ight)^{2} .$$
 (2.13)

Za r(t) vo vzťahu (2.13), predstavujúce skutočnú firing rate (ktorú nevieme presne vypočítať), dosádzame odhad firing rate, vypočítaný pomocou (2.5), o ktorom sme si povedali, že pri dostatočne veľa pokusoch je takmer totožný so skutočným r(t).

¹⁶ Môže to byť aj biely šum, prenásobený konštantou. To závisí od jednotiek parametra stimulu.

Teraz spočítame spike-triggered average na skutočných dátach. Budeme používať dáta z experimentu, publikovanom v [24], dostupné na [3]. Dáta pochádzajú z experimentu na muche (Calliphora vicina). Pri experimente sa púšťal stimul neurónu z V1, citlivému na vertikálny pohyb predmetu a zaznamenávali sa spiky. Dáta obsahujú jednotlivé časové kroky, dlhé 2 milisekundy, s informáciou o tom, či v nich nastal spike (0 a 1), a hodnota púšťaného stimulu¹⁷ v nich.

Nami vypočítaný spike-triggered average (pri $\tau_{\rm max} = 300 {\rm ms}$) vyzerá nasledovne:



Je dôležité poznamenať, že na grafoch na Obrázkoch 2.4 a 2.5 je časová os otočená. Je to spôsobené tým, že STA(τ) z (2.12) vyjadruje priemernú hodnotu stimulu v čase τ pred spikom. Väčšie hodnoty τ reprezentujú časy vzdialenejšie v minulosti od času spiku. Preto sa v neurovede využíva práve takéto vykreslenie STA, ktoré vizuálne viac korešponduje s časovým priebehom stimulu.

Na Obrázku 2.5 vidíme, že neurón na ktorom sa robil experiment, je citlivý na nárast hodnôt stimulu.

Je možné si všimnúť, že hodnoty STA, dostatočne vzdialené (do minulosti) od času spiku, sa pohybujú okolo nuly. Je to spôsobené tým, že takto vzdialené hodnoty stimulu nemajú vplyv na nastanie spiku, a preto hodnoty STA v takých

¹⁷ Hodnoty stimulu vyjadrujú relatívnu polohu predmetu na vertikálnej osi, ktorý sa ukazoval muche. Jednotky týchto hodnôt nie sú podstatné. Dôležité je len to, že čím sú vyššie hodnoty, tým vyššie je predmet a naopak.

časoch konvergujú k celkovému priemeru hodnôt púšťaného stimulu (čo je v prípade bieleho šumu 0).

Dalšia vec, ktorá stojí za povšimnutie, je, že pri hodnotách blízkych 0 STA prudko klesá. To by mohlo evokovať domnienku, že "náš" neurón nie je citlivý len na nárast hodnôt stimulu, ale na nárast nasledovaný prudkým poklesom. V skutočnosti je ale oveľa pravdepodobnejšie, že prudký pokles hodnôt STA v blízkosti 0¹⁸ je zapríčinený tým, že odpoveď neurónu (teraz pod ňou myslíme samotný spike) nejaký čas trvá, a preto jednotlivé spiky neodzrkadľujú odpoveď neurónu na stimul v časoch (0, ∞) pred spikom, ale v časoch (ε , ∞) pričom $\varepsilon > 0$.

Spomínané "oneskorenie" spôsobené trvaním spiku dĺžky ε nie je pre neurón vždy rovnaké a môže sa pri každom spiku trochu líšiť. Preto hodnoty STA neklesnú na hodnoty okolo 0 "skokom", ale klesnú spojito¹⁹.

2.3.3 Statická nelinearita

Modelovanie odpovede pomocou (2.11) má 2 problémy.

Prvým je, že odhadnutá odpoveď zo vzťahu (2.11) rastie nie len s rastúcou podobnosťou s a D, ale aj s rastúcimi absolútnymi hodnotami s (t.j. keď celý s prenásobíme konštantou, tou istou konštantou sa prenásobí aj výsledný odhad $\hat{r}(t)$). To je v spore s našim predpokladom, že neuróny sú citlivé najmä na určité zmeny v stimule, a nie na jeho absolútne hodnoty.

Druhým problémom modelu je fakt, že nijako nezaručuje, aby bola odpoveď nezáporná, čo je v priamom spore s Definíciou 1.

Tieto dva problémy sa v LNP modeli riešia pridaním tzv. *statickej nelinearity*. V praxi sa vzťah (2.11) zmení na:

$$\widehat{r}(t) = F\left(\int_0^{\tau_{\max}} s(t-\tau) D(\tau) d\tau\right).$$
(2.14)

Za F sa buď volia určité arbitrárne funkcie, alebo sa odhaduje z dát.

 $^{^{18}}$ V skutočnosti by bolo vše
obecnejšie namiesto "poklesu" k0 hovoriť o konvergenci
i k0, pretože STA môže mať aj záporné hodnoty.

¹⁹ Vzhľadom na spôsob vykreslenia grafu ako čiary spájajúcej body by však aj "skokový" pokles vyzeral "spojito".

Často sa volí napr. tangens hyperbolický, logistická funkcia alebo hocijaká iná rastúca funkcia, ktorá spĺňa podmienku nezápornosti a ohraničenosti (alebo aspoň konkávnosti).

Odhadovanie F z dát prebieha tak, že sa pustí stimul a vypočíta sa firing rate pomocou (2.5) a aj odhad pomocou lineárneho filtra (2.11). Následne sa ich hodnoty v rôznych časoch zakreslia ako je to na Obrázku 2.6, prebranom z [3].



X-ová os značí vypočítané hodnoty $r(t) \ge (2.5)$ a y-ová os odhady $\ge (2.11)$.

Na Obrázku 2.6 je vidieť, že medzi týmito hodnotami je istá závislosť. Krivka, ktorou sa dáta dajú "preložit", je dobrý odhad funkcie F.

Nevýhodou takéhoto odhadovania F je, že na odhad odpovede pre každý vybraný stimul (každý bod na grafe) pomocou (2.5) potrebujeme takýto stimul pustiť viackrát. Celkový počet pustení nejakého stimulu je potom veľmi vysoký a zaberá pri experimentálnych meraniach veľmi veľa času.

2.3.4 Možné problémy LNP modelu

LNP model predstavený v predošlej časti môže mať jeden zásadný problém. Ak by bol neurón citlivý na rôzne priebehy stimulu, napríklad na prudkú zmenu v stimule bez ohľadu na to, či by to bol pokles alebo nárast, STA (a teda aj $\hat{r}(t)$) by vyšiel približne nulový. V takomto prípade by bolo podľa [3] výhodnejšie mať viac kernelových funkcií *D*, každú prispôsobenú inému priebehu stimulu, na ktorý je neurón citlivý. Vzťah (2.14) by sa týmto zmenil na:

$$\hat{r}(t) = F\left(\int_0^{\tau_{\max}} s(t-\tau) D_1(\tau) d\tau, \int_0^{\tau_{\max}} s(t-\tau) D_2(\tau) d\tau, \dots, \int_0^{\tau_{\max}} s(t-\tau) D_m(\tau) d\tau\right). \quad (2.15)$$

Funkcia F by v tomto prípade bola funkciou $\mathbb{R}^m \to \mathbb{R}$, kde m vyjadruje počet rôznych priebehov stimulu, na ktoré je neurón citlivý.

Je zrejmé, že metódou STA opísanou v predošlej časti by sme jednotlivé D_i nenašli. Pokiaľ by sme však boli schopní rozdeliť jednotlivé časy spikov na viacero skupín podľa toho, aký typ priebehu stimulu ich spôsobil, mohli by sme na každej skupine vypočítať STA_i samostatne, a tak dostať D_i pre každú skupinu. Jednotlivé D_i by potom reprezentovali jednotlivé priebehy stimulov spôsobujúce spike.

Otázka je, ako rozdeliť jednotlivé časy spikov do skupín tak, aby v každej skupine boli časy spikov spôsobené tým istým priebehom stimulu. Jedným z možných riešení je nami vytvorený nasledovný algoritmus:

1. Časový úsek dĺžky τ_{max} pred každým spikom rozdelíme na η diskrétnych bodov. Priebeh stimulu na každom takomto časovom úseku bude reprezentovaný vektorom ($\tau_1, \tau_2, ..., \tau_\eta$), kde $\tau_{\text{max}} = \tau_\eta > ... > \tau_i = 0$. Zložky tohto vektora sú teda hodnoty stimulu v rôznych časoch pred daným spikom. 2. Vektory z predchádzajúceho kroku vykreslíme ako body v η – rozmernom priestore. Každý bod bude teda reprezentovať priebeh stimulu pred jedným konkrétnym spikom.

3. Na body z predchádzajúceho kroku aplikujeme nejaký zhlukovací (z angl. *cluster*) algoritmus, ktorým ich rozdelíme do zhlukov. Každý zhluk bude teda obsahovať podobné typy priebehu stimulu pred spikom.

4. Zo stimulov predchádzajúcim spikom, ktoré sú spolu (na základe predchádzajúceho kroku) v zhluku, vypočítame STA (a teda aj D) podľa (2.12). Takto vytvoríme m (m je počet zhlukov) funkcií D a následne firing rate odhadneme pomocou (2.15).

36
Tento algoritmus ukážeme na nami vygenerovaných ilustračných dátach. Budeme uvažovať neurón, ktorý je citlivý na prudký nárast, alebo prudký pokles stimulu.

Vygenerovali sme 1 000 000 bodový stimul ako biely šum. Každý bod reprezentuje časový úsek 10ms. Následne sme vygenerovali časy spikov tak, že v bodoch, ktorým predchádzal buď prudko rastúci alebo prudko klesajúci priebeh stimulu, nastávali spiky s výrazne vyššou pravdepodobnosťou. Informáciu o tom, či daný spike vznikol kvôli nárastu alebo kvôli poklesu stimulu, sme si ukladali. Spiky (a aj časti stimulov im predchádzajúce) sú teda vzhľadom na ich vznik dvoch typov.

Najprv sme vypočítali STA klasicky ako v časti 2.3.2, čiže sme "typy" spikov nijako nerozlišovali, pričom $\tau_{max} = 1$ s. Takto vypočítaný STA je na Obrázku 2.7. Tak, ako sme sa domnievali v prvom odseku tejto podkapitoly, STA je približne nulový. Nezodpovedá teda nijako priebehom stimulu, na ktorý je neurón citlivý



a LNP model by nefungoval správne.

Následne sme poslednú sekundu pred nastatím každého spiku rozdelili na $\eta = 100$ časových bodov,²⁰ pričom *i*-ty bod reprezentuje čas 1000 – 10**i* ms pred časom spiku. Pre každý spike sme takto dostali 100 zložkový vektor resp. bod v 100rozmernom priestore.

²⁰ Pri počítačovej simulácii je tento krok v podstate spravený už od začiatku, nakoľko pracujeme len s diskrétnymi hodnotami.

Na Obrázku 2.8 je zakreslených 1000 takýchto bodov. Vzhľadom na to, že 100-rozmerný priestor vykresliť nevieme, vykreslili sme iba 1. a 100. zložku každého vektora. Farebne sú rozlíšené spiky, ktoré vznikli z dôvodu prudkého poklesu stimulu (červená farba) a spiky, ktoré vznikli z dôvodu prudkého nárastu stimulu (modrá farba) – túto informáciu sme mali uloženú. Vidíme, že aj pri použití len dvoch



Obrázok 2.8 – prvá a posledná zložka bodov, zafarbená podľa toho, či príslušný spike nastal z dôvodu rastúceho alebo klesajúceho stimulu.

rozmerov (časov) sú body relatívne dobre odlíšiteľné. Sedí aj to, že modré body majú prvú zložku nízku a poslednú vysokú a červené majú prvú zložku vysokú a poslednú nízku.

Na tieto body sme následne aplikovali algoritmus *k-means*, publikovaný v [33]. Algoritmus k-means najprv vygeneruje *k* náhodných bodov, tzv. centroidov, pre predvolený počet *k* zhlukov. Príslušnosť bodu k zhluku je daná tým, ku ktorému centroidu je najbližšie. Algoritmus potom iteračne posúva centroidy tak, aby minimalizoval súčet štvorcov vzdialeností medzi bodmi a ich najbližšími centroidmi. Výhodou algoritmu je jeho jednoduchosť, nevýhodou je, že je až príliš závislý na vzdialenostiach jednotlivých bodov v priestore a nemá schopnosť "všimnúť si" zložitejšie štruktúry. Na náš účel je však ideálny.

Pri k-means musíme vždy preddefinovať počet zhlukov. Podľa [21] je na výber počtu zhlukov vhodné použiť tzv. lakťovú metódu (angl. *elbow method*). Lakťová metóda nám hovorí, že vhodný počet zhlukov je taký, pri ktorom sa "láme" pokles súčtu štvorcov vzdialeností bodov od centroidu zhluku. T.j. kedy už s pribúdajúcim počtom zhlukov tento súčet veľmi neklesá.

Výsledky lakťovej metódy môžeme vidieť na Obrázku 2.9. Vidíme, že pokles súčtu štvorcov vzdialeností bodov od centroidu zhluku je pri zmene z počtu zhlukov 1 na 2 veľký, ale pri zmene z 2 na 3 už až tak veľký nie je. Preto je vhodný počet zhlukov rovný 2. Tento počet je aj v súlade s faktom, že spiky sme generovali pri 2 rôznych priebehoch stimulu.

Je dôležité poznamenať, že informácia o tom, akého typu je daný bod (či reprezentuje klesajúci alebo rastúci priebeh stimulu, ktorý vyvolal spike) do zhlukovacieho algoritmu nijako nevstupuje.



Na Obrázku 2.10 na hornom grafe vidíme rovnaké vykreslenie bodov ako na Obrázku 2.8 s tým rozdielom, že body sú farebne rozdelené nie podľa informácie o tom, čo spôsobilo vygenerovanie spiku, ale podľa príslušnosti k zhluku. Vidíme, že je to veľmi podobné rozdeleniu na Obrázku 2.8. Potvrdzujú to aj výsledky algoritmu, kde je až 99% bodov správne²¹ zaradených v zhluku zodpovedajúcemu realite.

Na druhom grafe Obrázku 2.10 sú STA vypočítané zvlášť pre každý zhluk. Vidíme, že takto už jednotlivé STA zodpovedajú priebehu stimulu, na ktorý je neurón citlivý, a sú teda vhodná voľby pre funkcie *D*.

 $^{^{21}}$ Pod "správne" sa rozumie, že 99% bodov je v zhluku s tými bodmi, ktoré mali rovnaký "dôvod" vygenerovania spiku.



Obrázok 2.10 – Rozdelenie bodov algoritmom k
means a následné STA, vypočítane samostatne na každom zhluku.

V časti 2.3.4 sme teda ukázali, že pokiaľ máme neurón, ktorý je citlivý na rôzne typy priebehov stimulu, môžeme namiesto jedného lineárneho filtra použiť podľa [3] viacero filtrov. Navrhli sme aj možný spôsob, ako jednotlivé priebehy v dátach nájsť, a tak správne určiť počet a tvar filtrov.

2.3.5 Zhrnutie LNP modelu

Aj keď vzťahy (2.5) a (2.14) môžu pôsobiť len ako dve rôzne metódy odhadu odpovede, je dôležité si uvedomiť rozdiel medzi nimi.

Pri vzťahu (2.5) ide o odhadnutie firing rate zo spike trainov, ktoré vznikli z opakovania toho istého stimulu. Takto odhadnutú firing rate môžeme (pri dostatočnom množstve pokusov a dostatočne malom wt) pokladať (a v podkapitole 2.3 aj pokladáme) za skutočnú firing rate pre daný priebeh stimulu. Je to v podstate prechod medzi odpoveďou v podobe spike trainu a odpoveďou v podobe firing rate.

Pomocou LNP modelu odhadujeme všeobecný vzťah medzi stimulom a firing rate daného neurónu. Po vypočítaní STA z dát (natrénovaní modelu) vieme pri vhodnej voľbe nelineárnej funkcie F zo vzťahu (2.14) predikovať firing rate daného neurónu pre hocijaký nový priebeh stimulu. Následne z nej vieme generovať "Poissonovské" spike trainy tak, ako je popísané v 2.2.4.

Neurálne dekódovanie

3

V druhej kapitole sme sa zaoberali tvorbou modelu umožňujúceho predikovať odpoveď neurónu pri známom stimule.

Nervový systém ale čelí najmä opačnej úlohe. Dostane zakódovanú odpoveď od neurónov (v podobe spikov) a snaží sa odhadnúť, aký bol stimul, ktorý ju vyvolal. V tejto kapitole sa budeme zaoberať základnými modelmi takéhoto odhadovania, ktoré sú predmetom skúmania odvetvia neurovedy s názvom neurálne dekódovanie.

V celej kapitole budeme pod odpoveďou neurónu rozumieť počet spikov za jednotku času pri jednom pokuse a označovať ju písmenom r(zo slova "*response*"). Ide teda o hodnotu spike-count rate definovanú vzťahom (2.2) a nie o firing rate z Definície 1. Vo väčšine modelov, ktoré predstavíme, však nie je presná interpretácia odpovede dôležitá. Bude nás zaujímať hlavne to, že je to nejaká hodnota r (nezávislá od času), závislá na stimule s.

Púšťaný stimul bude vždy konštantný.

3.1 Vysvetlenie pravdepodobností

Ak predpokladáme, že stimul a odpoveď nadobúdajú len konečný počet diskrétnych hodnôt, dostávame nasledovné pravdepodobnosti:

- P(r) pravdepodobnosť, že odpoveď neurónu nadobudne hodnotu r,
- P(s) pravdepodobnosť, že stimul nadobudne hodnotu s,
- P(r,s) pravdepodobnosť, že odpoveď neurónu nadobudne hodnotu r a že stimul nadobudne hodnotu s,

• P(r|s) – pravdepodobnosť, že odpoveď neurónu nadobudne hodnotu r, ak stimul nadobudol hodnotu s,

• P(s|r) – pravdepodobnosť, že stimul nadobudol hodnotu s, ak odpoveď nadobudla hodnotu r.

Pri všetkých vyššie spomenutých pravdepodobnostiach by sa formálne malo namiesto r resp. s vo vnútri pravdepodobností písať R=r resp. S=s, kde R resp. Sreprezentuje označenie náhodnej premennej a r resp. s reprezentuje hodnotu tejto náhodnej premennej. Nakoľko je v neurovede zaužívané označovať tieto náhodné premenné a aj ich hodnoty malými písmenami, budeme v pravdepodobnostiach (a neskôr aj v hustotách pravdepodobnosti) písať vždy len jedno písmeno. T.j. P(r)bude znamenať pravdepodobnosť, že náhodná premenná odpoveď r nadobúda konkrétnu hodnotu r.

Pod S budeme rozumieť množinu všetkých možných stimulov a pod R množinu všetkých možných odpovedí. Pod "možnými" sa myslia tie, ktoré môžu nastať s nenulovou pravdepodobnosťou. Platia teda nasledovné vzťahy:

$$\sum_{R} P(r) = 1, \qquad \sum_{S} P(s) = 1,$$
 (3.1)

$$P(s) = \sum_{R} P(s|r)P(r), \qquad P(r) = \sum_{S} P(r|s)P(s), \qquad (3.2)$$

$$P(r,s) = P(r|s)P(s),$$
 $P(r,s) = P(s|r)P(r).$ (3.3)

Zo vzťahov (3.3) vyplýva tzv. Bayesova veta:

$$P(s|r) = \frac{P(r|s)P(s)}{P(r)}.$$
 (3.4)

Pre spojité hodnoty stimulu a odpovede budeme namiesto pravdepodobnosti P hovoriť o hustote pravdepodobnosti f a sumy vo vzťahoch (3.1) a (3.2) sa nahradia integrálmi.

3.2 Experiment s opicou

V tejto podkapitole ukážeme základný dekódovací model pre stimul s dvomi diskrétnymi hodnotami a spojitú odpoveď.

Budeme vychádzať z experimentu robeného s opicou v [1]. V experimente sa sledoval neurón opice citlivý na pohyb objektov smerom hore. Čím priamejšie nahor sa objekt (v tomto prípade bod) pohybuje, tým vyššia je aktivita takéhoto neurónu. Najvyššiu aktivitu má neurón vtedy, keď sa bod pohybuje priamo nahor a najnižšiu keď priamo nadol.



Obrázok 3.1 - Príklady pohybu bodov pri experimente pri rôznych koherenciách

Experiment pozostával z viacerých pokusov,²² pri ktorých opica sledovala pohyb skupiny bodov na obrazovke. V priebehu jedného pokusu sa smer pohybu jednotlivých bodov nemenil. Pokusy sa líšili v koherencii smerov pohybov jednotlivých bodov (Obrázok 3.1). Koherencia 0% znamená, že všetky body sa pohybujú náhodným smerom. Koherencia 50% znamená, že 50% bodov sa pohybuje vopred vybraným smerom (v tomto experimente buď priamo hore alebo priamo dole) a zvyšok sa pohybuje náhodne. Koherencia 100% znamená, že všetky body sa pohybujú jedným vopred vybraným smerom.

Pri každom pokuse sa zaznamenávala odpoveď neurónu r, teda počet spikov za jednotku času.

Na Obrázku 3.2 vidíme histogramy jednotlivých hodnôt odpovedí neurónu pri jednotlivých pokusoch a danej koherencii. Pre každú hodnotu koherencie máme dva farebne rozlíšené histogramy. Šrafovaním označené stĺpce zodpovedajú počtu pokusov s danou hodnotou odpovede, pri ktorých bol vybraný smer priamo nahor (budeme vtedy hovoriť, že stimul je "+") a čierne stĺpce vyjadrujú počet pokusov, kedy bol smer priamo nadol (budeme vtedy hovoriť, že stimul je "-")²³.

²² V práci budeme rozlišovať slová "experiment" (angl. *experiment*) a "pokus" (angl. *trial*). Pod "experimentom" budeme rozumieť celý experiment ako taký, zatiaľ čo pod "pokusom" jednu konkrétnu realizáciu daného experimentu (jedno meranie).

²³ Vo všeobecnosti sa pomocou znaku "+" označuje prítomnosť stimulu, na ktorý je neurón citlivý a znakom "–" neprítomnosť takého stimulu.



Obrázok 3.2 - Histogramy hodnôt odpovedí v jednotlivých pokusoch pri rôznych koherenciách

Prvý graf prislúcha koherencii 12.8%. To znamená, že 12.8% bodov sa pohybuje rovnakým, vopred vybraným smerom (v polovici pokusov kolmo nahor a v polovici kolmo nadol) a zvyšné sa pohybujú náhodne.

Vidíme, že pri vyššej koherencii bola odpoveď v prítomnosti stimulu + (šrafovaním označené histogramy) rádovo vyššia, ako pri stimule –. To len potvrdzuje, čo sme o neuróne už vedeli, že je citlivý na pohyb objektov smerom hore. Zvýšenie hodnoty odpovede neurónu teda indikuje, že stimul je +.

Čím nižšia koherencia (zvyšné grafy), tým vyšší je podiel bodov, ktoré sa pohybujú náhodne. Odpoveď preto nikdy nenadobúdala vysoké hodnoty. Strácal sa aj rozdiel vo veľkosti odpovedí pri pokusoch so stimulom + a stimulom –.

3.3 Dekódovací model

V predošlom experimente sa analyzovali odpovede neurónu na stimul + alebo – pri rôznej koherencii. Pri dekódovaní však vidíme iba odpoveď neurónu, zatiaľ čo stimul je neznámy. Chceli by sme preto zostrojiť model, ktorým budeme vedieť odhadnúť, o aký stimul ide, iba na základe odpovede, dát zozbieraných z predošlých pokusov a informácii o pravdepodobnostnom rozdelení možných hodnôt stimulu.

3.3.1 Optimálna rozhodovacia hranica

Budeme predpokladať, že náhodná premenná r má nejaké pravdepodobnostné rozdelenie za predpokladu, že je stimul + a iné pravdepodobnostné rozdelenie za predpokladu, že je stimul –. Hustoty pravdepodobností tejto premennej pri jednej resp. druhej podmienke budeme označovať ako $f_{r|s=+}(x)$ resp. $f_{r|s=-}(x)$. Vieme ich odhadnúť pomocou zozbieraných dát (napr. pomocou histogramov z Obrázku 3.2).

Jednoduchý prístup ako určiť na základe nameranej odpovede a hustôt $f_{r|s}(x)$, či ide o stimul + alebo –, je zvoliť vhodnú hranicu z s tým, že ak bude r > z, tak označíme stimul ako + a inak ako –. Tento prístup vyplýva z predpokladu, že neurón preferuje stimul + pred –, a teda má v jeho prítomnosti zvýšenú hodnotu odpovede r (tak ako to bolo v prípade neurónu z podkapitoly 3.2). Tento prístup sa používa aj v [22].

Je zrejmé, že ak bude stimul+,tak ho pri takom
to prístupe správne označíme s pravdepodobnosťou:

$$P(r > z \mid s = +) = \int_{z}^{\infty} f_{r|s=+}(x) dx = \beta(z)$$
(3.5)

a ak bude stimul –, tak ho nesprávne označíme za stimul + s pravdepodobnosťou:

$$P(r > z \mid s = -) = \int_{z}^{\infty} f_{r \mid s = -}(x) dx = a(z).$$
 (3.6)

Zároveň, ak bude stimul – tak ho správne označíme s pravdepodobnosťou:

$$P(r < z \mid s = -) = \int_{-\infty}^{z} f_{r \mid s = -}(x) dx = 1 - a(z)$$
(3.7)

a ak bude stimul+,tak ho nesprávne označíme za stimul-s pravdepodobnosťou:

$$P(r < z \mid s = +) = \int_{-\infty}^{z} f_{r \mid s = +}(x) dx = 1 - \beta(z).$$
 (3.8)

Z uvedeného vyplýva, že pravdepodobnosť správneho určenia stimulu je:

$$P(\text{správne}) = \beta(z)P(s=+) + (1-a(z))P(s=-).$$
(3.9)

Konštanty P(s = +) a P(s = -) sú všeobecné pravdepodobnosti výskytu stimulu + resp. – (v pokuse v minulej podkapitole sa pri polovici pokusov volil stimul + a v polovici –, čiže obe tieto pravdepodobnosti boli rovné 1/2). Naviac ešte platí:

$$\int_{-\infty}^{\infty} f_{r|s=-}(x) dx = \int_{-\infty}^{\infty} f_{r|s=+}(x) dx = \int_{-\infty}^{\infty} f_r(x) dx = 1$$
(3.10)

 \mathbf{a}

$$P(s = -)f_{r|s=-}(x) + P(s = +)f_{r|s=+}(x) = f_r(x)$$
(3.11)

(3.10) vyplýva z faktu, že sú to hustoty a (3.11) z faktu, že stimul v našom prípade nadobúda len 2 hodnoty.

Teraz odvodíme optimálnu hodnotu hranice z. Bude to hodnota, ktorá maximalizuje pravdepodobnosť správneho určenia stimulu pri danej odpovedi r. Budeme predpokladať, že náhodná premenná r|s=+ má rozdelenie $N(\mu_+, \sigma_+^2)$ a náhodná premenná r|s=- má rozdelenie $N(\mu_-, \sigma_-^2)^{24}$ čiže $f_{r|s=+}(x) = \frac{1}{\sqrt{2\pi\sigma_+^2}}e^{-\frac{(x-\mu_+)^2}{2\sigma_+^2}}$

a
$$f_{r|s=-}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu_{-})^2}{2\sigma^2}}$$
. Po dosadení do (3.5, 3.7) a do (3.9) dostávame:

$$P(\text{správne}) = P(s = +) \int_{z}^{\infty} \frac{1}{\sqrt{2\pi\sigma_{+}^{2}}} e^{-\frac{(x-\mu_{+})^{2}}{2\sigma_{+}^{2}}} dx + P(s = -) \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi\sigma_{-}^{2}}} e^{-\frac{(x-\mu_{-})^{2}}{2\sigma_{-}^{2}}} dx.$$
(3.12)

Túto pravdepodobnosť chceme maximalizovať vzhľadom na z, preto ju zderivujeme:

$$\begin{aligned} \frac{dP(\operatorname{správne})}{dz} &= \\ &= P(s=+)\frac{d}{dz} \int_{z}^{\infty} \frac{1}{\sqrt{2\pi\sigma_{+}^{2}}} e^{-\frac{(x-\mu_{+})^{2}}{2\sigma_{+}^{2}}} dx + P(s=-)\frac{d}{dz} \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi\sigma_{-}^{2}}} e^{-\frac{(x-\mu_{-})^{2}}{2\sigma_{-}^{2}}} dx = \\ &= P(s=+)\frac{-1}{\sqrt{2\pi\sigma_{+}^{2}}} e^{-\frac{(z-\mu_{+})^{2}}{2\sigma_{+}^{2}}} + P(s=-)\frac{1}{\sqrt{2\pi\sigma_{-}^{2}}} e^{-\frac{(z-\mu_{-})^{2}}{2\sigma_{-}^{2}}} \end{aligned}$$

a položíme rovnú 0:

$$0 = \frac{dP(\text{správne})}{dz}$$

 $^{^{24}}$ V literatúre sa často používa tento predpoklad. Nie je však nutný, pretože model s hranicou z platí aj pri iných rozdeleniach. Experimenty (ako napr aj 3.2) však naznačujú, že odpoveď neurónu má naozaj často približne normálne rozdelenie.

$$\begin{split} 0 &= P(s=+) \frac{-1}{\sqrt{2\pi\sigma_+^2}} e^{-\frac{(z-\mu_+)^2}{2\sigma_+^2}} + P(s=-) \frac{1}{\sqrt{2\pi\sigma_-^2}} e^{-\frac{(z-\mu_-)^2}{2\sigma_-^2}} \\ &= \frac{e^{-\frac{(z-\mu_+)^2}{2\sigma_+^2}}}{e^{-\frac{(z-\mu_+)^2}{2\sigma_-^2}}} = \frac{\sqrt{2\pi\sigma_+^2}}{\sqrt{2\pi\sigma_-^2}} \frac{P(s=-)}{P(s=+)} \\ &- \frac{(z-\mu_+)^2}{2\sigma_+^2} + \frac{(z-\mu_-)^2}{2\sigma_-^2} = \ln\left(\frac{\sqrt{2\pi\sigma_+^2}}{\sqrt{2\pi\sigma_-^2}} \frac{P(s=-)}{P(s=+)}\right) \\ 0 &= (-\sigma_-^2 + \sigma_+^2)z^2 + (2\sigma_-^2\mu_+ - 2\sigma_+^2\mu_-)z - \sigma_-^2\mu_+^2 + \sigma_+^2\mu_-^2 - 2\sigma_+^2\sigma_-^2 \ln\left(\frac{\sqrt{2\pi\sigma_+^2}}{\sqrt{2\pi\sigma_-^2}} \frac{P(s=-)}{P(s=+)}\right) \end{split}$$

Dostávame tak kvadratickú rovnicu s 2 koreňmi:

$$z_{1,2} = \frac{-b \mp \sqrt{b^2 - 4ac}}{2a},$$
(3.13)

kde

$$egin{aligned} a &= (-\sigma_{-}^2 + \sigma_{+}^2),\ b &= (2\sigma_{-}^2\mu_{+}^2 - 2\sigma_{+}^2\mu_{-}^2) ext{ a}\ c &= -\sigma_{-}^2\mu_{+}^2 + \sigma_{+}^2\mu_{-}^2 - 2\sigma_{+}^2\sigma_{-}^2\ln\!\left(\!rac{\sqrt{2\pi\sigma_{+}^2}}{\sqrt{2\pi\sigma_{-}^2}}rac{P(s=-)}{P(s=+)}\!
ight)\!. \end{aligned}$$

Na záver z koreňov z_1 a z_2 vyberieme taký, pre ktorý je hodnota P(správne) vyššia. Pokiaľ platí $\sigma_{-}^2 = \sigma_{+}^2$ máme iba lineárnu rovnicu a z zvolíme ako z = -c/b.

Na Obrázku 3.3 môžeme vidieť ilustračný príklad neurónu, ktorého odpoveď r má za podmienky s=+ rozdelenie $N(4, 1^2)$ a za podmienky s=- má rozdelenie $N(2, 0.8^2)$. Stimul + má pravdepodobnosť P(s=+) = 1/4 a stimul – má pravdepodobnosť P(s=-) = 3/4.

Na grafe je zakreslená hustota $f_{r|s=+}(x)$ prenásobená P(s = +) a hustota $f_{r|s=-}(x)$ prenásobená P(s = -). Je tam aj zaznačená hranica z, vypočítaná pomocou vzťahu (3.13). Môžeme si všimnúť, že hranica je presne v mieste, kde sa červená krivka s modrou pretínajú. Toto nie je náhoda. Hranica z predsa hovorí o tom, od ako vysokej odpovede r máme označovať stimul ako +. Je intuitívne zrejmé, že



Obrázok 3.3 - Príklad hustôt pravdepodobnostných rozdelení odpoved
e v závislosti od stimulu a optimálna hranica z

v oblasti, kde je červená krivka vyššia, je aj vyššia pravdepodobnosť, že daná odpoveď bude prislúchať stimulu + a v oblasti, kde je vyššia modrá krivka, je vyššia pravdepodobnosť, že daná odpoveď bude prislúchať stimulu -. Toto intuitívne tvrdenie teda presne korešponduje s vypočítanou optimálnou hodnotou z.

Pri neuróne z Obrázku 3.3 a našom dekódovacom modeli je pravdepodobnosť správneho určenia stimulu (podľa vzťahu (3.12)) približne 90%. Je dobré si uvedomiť, že čím menej sa hustoty (červená a modrá krivka) prekrývajú, tým vyššia je táto pravdepodobnosť. Tento fakt bol naznačený už v podkapitole 3.2, kde sa na Obrázku 3.2 so znižujúcou koherenciou histogramy jednotlivých stimulov a odpovedí k sebe približovali a so zvyšujúcou vzďaľovali.

3.3.2 ROC krivka

Na analýzu závislosti hodnôt $\beta(z)$ a a(z) (pozri str. 46) od hranice z sa dá použiť tzv. ROC krivka (angl. reciever operating characteristics curve). ROC krivka je tvorená bodmi, z ktorých každý prislúcha nejakej hodnote z. Súradnica bodu na osi x je a(z) a na osi y je $\beta(z)$.

Bod ROC krivky prislúchajúci z = 0 má súradnice (1,1), pretože r > 0 platí vždy (a teda $P(r > z \mid s = +)$ aj $P(r > z \mid s = -)$ sú 1). V opačnom prípade, pre body ROC krivky prislúchajúce hodnotám z vyšším ako r_{max}^{25} , nebude r > z platiť nikdy, a preto $P(r > z \mid s = +)$ aj $P(r > z \mid s = -)$ budú 0.

Nakoľko $\beta(z)$ je pravdepodobnosť správneho určenia stimulu + a a(z) je pravdepodobnosť nesprávneho označenia stimulu – za stimul +, chceme, aby $\beta(z)$ bolo čo najvyššia a a(z) čo najnižšia. Preto hľadáme takú hodnotu z, pri ktorej je bod ROC krivky čo najbližšie k bodu (0,1).

Pokiaľ sa prioritne chceme vyhnúť chybe prvého druhu (označenie stimulu – ako stimul +), posunieme hranicu z vyššie a naopak, pokiaľ sa prioritne chceme vyhnúť chybe druhého druhu (neoznačenie stimulu + ako stimul +), tak posunieme hranicu z nižšie. To má zmysel vtedy, pokiaľ je jedna z chýb (prvého alebo druhého druhu) horšia, ako druhá (viac v časti 3.3.4).

3.3.3 Generovanie ROC kriviek a numerické počítanie optimálnej hranice na základe rôznych hustôt

V časti 3.3.1 sme ukázali analytický výpočet optimálnej hranice z. Výpočet spočíval v hľadaní maxima funkcie P(správne). Pri maximalizovaní tejto funkcie však potrebujeme poznať hustotu $f_{r|s=+}(x)$. Tá v realite často známa nie je, známe sú iba dáta, t.j. jednotlivé realizácie náhodnej premennej r získané z jednotlivých pokusoch. S týmto faktom sa dá vysporiadať dvoma spôsobmi:

1. Odhadnúť funkciu hustoty na základe dát a vypočítať z analyticky tak, ako v 3.3.1.

2. Vypočítať z z dát numericky.

V nasledujúcej časti tejto podkapitoly sú prezentované výsledky numerického počítania hranice z a ROC krivky z nami vygenerovaných ilustračných dát.

Dáta budeme generovať tak, aby odzrkadľovali správanie sa rôznych typov neurónov spomenutých v [13]. Pod typom neurónu sa v tomto prípade rozumie to, aké má rozdelenie jeho odpoveď v prítomnosti stimulu +.

²⁵ Neurón má obmedzený počet spikov, ktoré je schopný za určitý čas vygenerovať. r_{max} je konštanta – maximálna možná odpoveď neurónu pre daný experiment (z biofyzikálnych dôvodov).

Uvedieme príklady 4 neurónov, každý s inou hustotou premennej r|s=+. V niektorých príkladoch budeme uvažovať aj iné hustoty ako je hustota normálneho rozdelenia. Hustota $f_{r|s=-}(x)$ premennej r|s=- (teda odpovede neurónu pri neprítomnosti stimulu) bude vo všetkých príkladoch rovnaká.

Pre každý príklad neurónu (resp. pre vygenerované hodnoty jeho odpovede r v prítomnosti a aj v neprítomnosti stimulu) vypočítame príslušnú ROC krivku.

Na Obrázku 3.4 každý riadok zodpovedá inému príkladu neurónu. Vľavo sú vykreslené funkcie hustôt premenných r|s=-a r|s=+, z ktorých sme na základe Vety 4.1 z [8] generovali dáta²⁶.

Na generovanie dát sme použili postupne nasledovné hustoty:

$$f_{r|s=-}(x) = f_{-}(x) = \frac{1}{0.2\sqrt{2\pi}} e^{\frac{-(x-1)^2}{2 \cdot 0.2^2}},$$
 (3.14)

$$f_{r|s=+}(x) = f_{+1}(x) = \frac{2}{0.2 \cdot 3\sqrt{2\pi}} e^{\frac{-(x-1)^2}{2 \cdot 0.2^2}} + \frac{1}{0.2 \cdot 3\sqrt{2\pi}} e^{\frac{-(x-2)^2}{2 \cdot 0.2^2}},$$
(3.15)

$$f_{r|s=+}(x) = f_{+_2}(x) = \frac{2}{0.2 \cdot 3\sqrt{2\pi}} e^{\frac{-(x-1.2)^2}{2 \cdot 0.2^2}} + \frac{1}{0.2 \cdot 3\sqrt{2\pi}} e^{\frac{-(x-2)^2}{2 \cdot 0.2^2}},$$
(3.16)

$$f_{r|s=+}(x) = f_{+_3}(x) = \frac{1}{0.3\sqrt{2\pi}} e^{\frac{-(x-1.3)^2}{2\cdot 0.3^2}},$$
(3.17)

$$f_{r|s=+}(x) = f_{+4}(x) = 0.8 \frac{1}{0.06\sqrt{2\pi}} e^{\frac{-(x-1)^2}{2 \cdot 0.06^2}} + 0.2 \frac{1}{0.2\sqrt{2\pi}} e^{\frac{-(x-1.4)^2}{2 \cdot 0.2^2}}.$$
 (3.18)

V strednom stĺpci sú zobrazené histogramy takto vygenerovaných dát. Zakreslená je aj optimálna hranica z, vypočítaná numericky z vygenerovaných dát. Spôsob jej výpočtu prebiehal tak, že sme os r rozdelili na diskrétne body a pre každý bod sme počítali počet realizácii premennej r, vygenerovaných pri stimule +, s vyššou hodnotou ako daný bod a k tomuto počtu sme pripočítali počet realizácii premennej r, vygenerovaných pri stimule –, s nižšou hodnotou ako daný bod. Následne sme vybrali taký bod, pre ktorý bol tento súčet najvyšší a prehlásili ho za optimálnu

²⁶ Ako funkciu h sme použili hustotu rozdelenia R(-1,4). Uvedomujeme si, že predpoklad spomínanej vety, aby boli hustoty f a h kladné na rovnakom intervale a inde nulové, splnený nie je, ale všetky hustoty, z ktorých dáta generujeme (čiže funkcie f zo spomínanej vety), majú mimo intervalu (-1,4) tak nízke hodnoty, že korektnosť generovania týmto spôsobom nie je nijako narušená.



hranicu z. Znova sa potvrdzuje to, čo napovedá intuícia (a to čo nám vyšlo aj pri analytickom riešení v 3.5), že optimálna hranica z je na priesečníku hustôt.

Obrázok 3.4 - Nami vygenerované dáta na základe zvolených hustôt a k nim vypočítané ROC krivky. Pozor! Vo štvrtom riadku na ľavom a strednom obrázku je iná mierka osi y ako na ostatných obrázkoch.

Vo štvrtom príklade máme hustoty, ktoré majú viacero priesečníkov. Aj v tomto prípade platí, že optimálna hranica z je na priesečníku. Avšak to, na ktorom konkrétne, je dané až tvarom samotných hustôt, z ktorých generujeme dáta, a teda samotnými dátami.

V pravej časti sú zakreslené ROC krivky, vypočítané z dát, so zaznačeným bodom, prislúchajúcim optimálnej hodnote z. V prvých dvoch príkladoch máme hustoty, pre ktoré ROC krivka najprv prudko stúpa (zvyšuje sa $\beta(z)$ bez toho, aby sa nejako výrazne zvýšila a(z)). To je spôsobené tým, že časť pokusov s vysokou hodnotou r sa dá veľmi jednoznačne určiť ako pokusy, v ktorých bol prítomný stimul +. Neodborne povedané, v pravej časti, kde pravý "kopček" hustoty $f_{+_1}(x)$ resp. $f_{+_2}(x)$ nadobúda relatívne vysoké hodnoty, je $f_{-}(x)$ takmer nulová).

V prípade hustoty $f_{+_1}(x)$ (prvý príklad) má jej ľavý "kopček" rovnakú strednú hodnotu ako hustota premennej $f_{-}(x)$. To spôsobuje, že ROC krivka je od istej hodnoty z lineárna, so sklonom rovným pomeru veľkostí "kopčekov".

Pri hustotách zo štvrtého príkladu vidíme, že pri optimálnej hodnote $z \, \text{sú} \, \beta(z)$ aj a(z) relatívne vysoké. To znamená, že pravdepodobnosť správneho určenia stimulu + je vysoká, ale tak isto je aj vysoká pravdepodobnosť nesprávneho označenia stimulu – za stimul +. To je spôsobené tým, že optimálna hranica z je príliš nízka, a preto takmer každý stimul označujeme ako +.

Pri hustote z prvého príkladu nastáva zase opačný prípad. Hodnota z je relatívne vysoká, a preto väčšinu stimulov označujeme ako –. To sa na ROC krivke prejavuje tak, že pri optimálnej hodnote z sú hodnoty $\beta(z)$ a a(z) relatívne nízke.

Na záver ešte uvedieme pravdepodobnosti P(správne) pre jednotlivé príklady vypočítané numericky z vygenerovaných dát pri optimálnom z. Pravdepodobnosti P(s = +) a P(s = -) boli obe rovné 1/2, čo znamená, že v každom príklade sme polovicu dát generovali z hustoty $f_{-}(x)$ a polovicu z $f_{+_i}(x)$. Pre prvý príklad je $P_1(\text{správne}) \approx 66.5\%$, pre druhý $P_2(\text{správne}) \approx 76\%$, pre tretí $P_3(\text{správne}) \approx 74\%$ a pre štvrtý $P_4(\text{správne}) \approx 62\%$. Opäť sa ukazuje, že čím sú hustoty $f_{-}(x)$ a $f_{+}(x)$ "rozdielnejšie" (t.j. majú menší prienik), tým je pravdepodobnosť uhádnutia pri našom modeli s optimálnou hranicou vyššia.

3.3.4 Strata za chybu

V predchádzajúcich častiach tejto podkapitoly sme predstavili model, ktorý maximalizuje pravdepodobnosť správneho určenia stimulu, resp. minimalizuje pravdepodobnosť chyby. Spomenuli sme aj, že exitujú dva typy chýb: chyba prvého druhu (označenie stimulu – ako stimul +) a chyba druhého druhu (neoznačenie stimulu + ako stimul +). Straty za tieto chyby ale nie sú vždy rovnaké.

Napríklad, ak vidí mucha objekt pohybujúci sa vertikálne dole k nej, tak strata za chybu prvého druhu je len zbytočná strata energie, zatiaľ čo strata za chybu druhého druhu je smrť.

Ak označíme stratu za chybu prvého druhu ako L(+) (lebo vtedy chybne označujeme stimul ako +) a stratu za chybu druhého druhu ako L(-), tak vieme preformulovať vzťah (3.9) na celkovú stratu za chybu:

$$L = (1 - \beta(z))P(s = +)L(-) + a(z)P(s = -)L(+).$$
(3.19)

Nakoľko L(+) a L(-) sú konštanty, L vieme minimalizovať rovnakým spôsobom ako sme vo vzťahu (3.9) maximalizovali P(správne).

3.4 Dekódovanie populácie neurónov

Na záver kapitoly 3 ukážeme dekódovací model, ktorý sa zaoberá nie jedným, ale viacerými (N) neurónmi pričom predpokladáme, že každý z nich generuje spiky nezávisle od ostatných. Budeme uvažovať vektor $r = (r_1, r_2, ..., r_N)$, ktorého zložkami sú odpovede jednotlivých neurónov, s konkrétnou realizáciou označenou ako $r_0 = (r_{0_1}, r_{0_2}, ..., r_{0_N})$. Budeme hľadať takú hodnotu stimulu s, pri ktorej je takýto vektor odpovedí r_0 najpravdepodobnejší. Budeme vychádzať zo spojitej verzie Bayesovej vety (3.4):

$$f(s|r) = \frac{f(r|s)f(s)}{f(r)}.$$
(3.20)

Hustota f(s|r) sa nazýva aposteriórna (z lat. a posteriori) a vyjadruje pravdepodobnostné rozdelenie stimulu v závislosti od odpovede. Hustota f(r|s) sa nazýva funkcia vierohodnosti (angl. *likelihood*) a vyjadruje pravdepodobnostné rozdelenie odpovede v závislosti od stimulu. Hustota f(s) sa nazýva apriórna (z lat. a priori) a vyjadruje rozdelenie stimulu ako takého (výskyt daných javov alebo situácií v prírode). Hustota f(r) sa nazýva marginálna a vyjadruje rozdelenie odpovede bez ohľadu na stimul, je to všeobecná vlastnosť neurónu, v akej intenzite väčšinou generuje/je schopný generovať spiky.

Práve funkcie f(s|r) a f(r|s) budeme v tejto podkapitole maximalizovať cez s pre fixné $r = r_0$.

Je dôležité uvedomiť si, že stimul vždy (časovo) predchádza odpovedi bez ohľadu na to, či ide o f(s|r) alebo o f(r|s).

V prípade f(s|r) sa pri fixnom $r = r_0$ s meniacim sa *s* nemení graf funkcie $f(s|r=r_0)$, ale mení sa hodnota na horizontálnej *s*-osi. Hľadáme teda takú hodnotu $s=s_{MAP}$, pre ktorú má hustota $f(s|r=r_0)$ najvyššiu hodnotu. Naopak, pri $f(r=r_0|s)$ sa nemení hodnota na horizontálnej *r*-osi, ale mení sa graf funkcie $f(r=r_0|s)$ (lebo ten je v tomto prípade daný premennou *s*). Hľadáme teda takú funkciu, danú hodnotou $s=s_{ML}$, ktorá má v bode r_0 najvyššiu hodnotu.

Najprv budeme hľadať ML (maximum likelyhood) odhad, čiže $s_{\rm ML}$. Je to stimul, ktorý maximalizuje $f(r=r_0|s)$. Ľudskou rečou vieme stimul $s_{\rm ML}$ interpretovať ako odpoveď na otázku: "Pri akom stimule bola nameraná odpoveď r_0 najpravdepodobnejšia".

Keďže odpoveď r_a (*a*-ta zložka vektora r) predstavuje počet spikov *a*-teho neurónu za jednotku času v pokuse (zo vzťahu (2.2): $r_a = n_a/T$), tak nadobúda len diskrétne hodnoty, pretože počet spikov *a*-teho neurónu v pokuse n_a je diskrétna veličina a čas trvania pokusu T je konštanta. Budeme teda namiesto $f(r=r_0|s)$ maximalizovať $P(r=r_0|s)$. Náhodné premenné r_a a n_a sú vzhľadom na to, že T je konštanta, úplne závislé, a preto platí P(r|s) = P(n|s). Zo vzťahu (2.7) dostávame pre *a*-ty neurón:

$$P(r_a = r_{0_a}|s) = P(n_a = n_{0_a}|s) = \frac{(\varphi_a(s)T)^{n_{0_a}}}{n_{0_a}!} e^{-\varphi_a(s)T} = \frac{(\varphi_a(s)T)^{(r_{0_a}T)}}{(r_{0_a}T)!} e^{-\varphi_a(s)T}.$$
 (3.21)

Vo vzťahu (2.7) sme na vyjadrenie pravdepodobnosti *n* spikov používali konštantnú firing rate. V tretej kapitole síce firing rate nepočítame (počítame iba počet spikov za jednotku času v pokuse), ale to však nevadí, pretože konštantná firing rate je (logicky) rovná priemernej firing rate a priemernú firing rate vieme (pozri str. 24) vyjadriť pomocou tunning curve $\varphi_a(s)$.

Pre a-ty neurón budeme uvažovať nasledovný tvar tunning curve:

$$r_a = \varphi_a(s) = r_{\max} e^{-rac{1}{2} rac{(s-s_a)^2}{\sigma_a}}.$$
 (3.22)

Kde s_a je preferovaný stimul *a*-teho neurónu a σ_a je rozptyl jeho odpovedí. Naviac budeme o tunning curves jednotlivých neurónov predpokladať, že ich súčet je konštantný vzhľadom na s. Čiže $\sum_{a=1}^{N} \varphi_a(s)$ je konštanta. Príklad takýchto tunning curves, rovnomerne rozložených vzhľadom na s, je ilustrovaný na Obrázku 3.5. Takýto príklad tunning curve aj so spomenutým predpokladom a obrázkom je uvedený v [3].



Obrázok 3.5 – Príklad rovnomerne rozmiestnených tunning curves

Keďže sú odpovede neurónov nezávislé, môžeme napísať:

$$egin{aligned} P(r=r_0 \mid s) &= Pig(r_1 = r_{0_1} \land ... \land r_{
m N} = r_{0_{
m N}} \mid sig) = Pig(r_1 = r_{0_1} \mid sig). \ ... \ .Pig(r_N = r_{0_N} \mid sig) = \ &= \prod_{a=1}^N Pig(r_a = r_{0_a} \mid sig) = \prod_{a=1}^N rac{(arphi_a(s)\,T)^{(r_{0_a}T)}}{(r_{0_a}T)!} {
m e}^{-arphi_a(s)T}. \end{aligned}$$

Vzhľadom na to, že logaritmus je rastúca funkcia, tak $s_{\rm ML}$, ktoré maximalizuje výraz $P(r = r_0 | s)$, je to isté ako $s_{\rm ML}$, ktoré maximalizuje výraz $\ln(P(r = r_0 | s))$. Preto budeme hľadať argument maxima výrazu $\ln(P(r = r_0 | s))$. Zderivujeme podľa s a položíme rovné 0:

$$egin{aligned} 0 &= rac{d}{ds} \mathrm{ln} \left(P(r = r_0 \mid s)
ight) \,= \ &= rac{d}{ds} igg(\sum_{a=1}^N \left(r_{0_a} T(\ln \left(arphi_a(s)
ight) + \ln \left(T
ight)
ight) - \ln \left(\left(r_{0_a} T
ight)!
ight) - arphi_a(s) T
ight) igg). \end{aligned}$$

Všetky členy, ktoré neobsahujú s vypadnú, tak isto vypadne aj člen $\frac{d}{ds} \left(\sum_{a=1}^{N} (\varphi_{a}(s)T) \right), \text{ pretože o } \sum_{a=1}^{N} (\varphi_{a}(s)) \text{ predpokladáme, že je konštanta.}$ Dostávame:

$$0 = \frac{d}{ds} \left(\sum_{a=1}^{N} \left(r_{0_a} T \ln \left(\varphi_a(s) \right) \right) \right) = \sum_{a=1}^{N} \left(r_{0_a} T \frac{\varphi'_a(s_{\mathrm{ML}})}{\varphi_a(s_{\mathrm{ML}})} \right)$$

Po dosadení za $\varphi_a(s)$ z (3.22) dostávame:

$$0 = \sum_{a=1}^{N} \left(r_{0_a} T \frac{(s_{\rm ML} - s_a)}{\sigma_a} \right)$$

$$0 = \sum_{a=1}^{N} \left(\frac{(r_{0_a} s_{\rm ML} - r_{0_a} s_a)}{\sigma_a} \right)$$

$$0 = s_{\rm ML} \sum_{a=1}^{N} \left(\frac{r_{0_a}}{\sigma_a} \right) - \sum_{a=1}^{N} \left(\frac{r_{0_a} s_a}{\sigma_a} \right)$$

$$s_{\rm ML} = \frac{\sum_{a=1}^{N} \left(\frac{r_{0_a} s_a}{\sigma_a} \right)}{\sum_{a=1}^{N} \left(\frac{r_{0_a}}{\sigma_a} \right)}.$$
(3.23)

Našli sme teda stimul $s_{\rm ML},$ pri ktorom má nameraná odpoveď najvyššiu pravdepodobnosť.

Zo vzťahu (3.23) vidíme, že čím vyššie je r_{0_a} , tým väčší vplyv má s_a na $s_{\rm ML}$. To je logické, pretože r_{0_a} nadobúda vysoké hodnoty práve vtedy, keď je púšťaný stimul podobný preferovanému stimulu *a*-teho neurónu, čiže s_a .

Ďalšia vec, ktorú môžeme pozorovať, je, že ak je miera variability odpovedí $\sigma_a a$ -teho neurónu vysoká, r_{0_a} ani s_a nemajú veľký vplyv na $s_{\rm ML}$. To je tiež logické, keďže odpoveď neurónu s vysokou variabilitou odpovedí nemá pre nás veľmi výpovednú hodnotu.

Teraz vypočítame hodnotu $s=s_{MAP}$. To je stimul, ktorý maximalizuje $f(s|r=r_0)$. Ľudskou rečou je to odpoveď na otázku: "Aký stimul bol vzhľadom na nameranú odpoveď najpravdepodobnejší?".

Výpočet (aj so všetkými úvahami a predpokladmi) bude rovnaký ako pre $s_{\rm ML}$, ale na vyjadrenie $f(s|r=r_0)$ pomocou $P(r_a=r_{0_a}|s)$ použijeme Bayesovu vetu (pre diskrétne r a spojité s):

$$f(s|r) = \frac{P(r|s)f(s)}{P(r)}.$$
 (3.24)

Dostávame teda:

$$0 = \frac{d}{ds} \ln (f(s|r)) =$$

$$= \frac{d}{ds} (\ln (P(r = r_0 | s)) + \ln(f(s)) - \ln(P(r = r_0)) =$$

$$= \sum_{a=1}^{N} \left(r_{0_a} T \frac{(s_{\text{MAP}} - s_a)}{\sigma_a} \right) + \frac{f'(s_{\text{ML}})}{f(s_{\text{ML}})} =$$

$$= \sum_{a=1}^{N} \left(r_{0_a} T \frac{(s_{\text{MAP}} - s_a)}{\sigma_a} \right) + \frac{s_{\text{MAP}} - s_{\text{prior}}}{\sigma_{\text{prior}}} =$$

$$= Ts_{\text{MAP}} \sum_{a=1}^{N} \left(\frac{r_{0_a}}{\sigma_a} \right) - T \sum_{a=1}^{N} \left(\frac{r_{0_a} s_a}{\sigma_a} \right) + \frac{s_{\text{MAP}} - s_{\text{prior}}}{\sigma_{\text{prior}}} =$$

$$= s_{\text{MAP}} \left(T \sum_{a=1}^{N} \left(\frac{r_{0_a}}{\sigma_a} \right) + \frac{1}{\sigma_{\text{prior}}} \right) - \left(T \sum_{a=1}^{N} \left(\frac{r_{0_a} s_a}{\sigma_a} \right) + \frac{s_{\text{prior}}}{\sigma_{\text{prior}}} \right) \Rightarrow$$

$$\Rightarrow s_{\text{MAP}} = \frac{T \sum_{a=1}^{N} \left(\frac{r_{0_a} s_a}{\sigma_a} \right) + \frac{s_{\text{prior}}}{\sigma_{\text{prior}}}.$$
(3.25)

Pri štvrtej rovnosti predpokladáme, že apriórna hustota f(s) má normálne rozdelenie so strednou hodnotu s_{prior} a varianciou σ_{prior} . Je vidieť, že čím je nižšia variancia stimulu σ_{prior} , tým väčší vplyv má jeho stredná hodnota s_{prior} na s_{MAP} . Je to spôsobené tým, že čím nižšia je variancia, tým jednoznačnejšie vieme povedať, aké hodnoty stimul zvykne nadobúdať a aké sú teda najpravdepodobnejšie.

Spôsob, akým sme odvodili $s_{\rm MAP}$
a $s_{\rm ML}$ je načrtnutý v [5].

4 Neurónové siete

Neurónov máme v tele podľa [20] okolo 86 miliárd a spojení medzi nimi 10 000 krát toľko. Na lepšie pochopenie toho, ako nervová sústava funguje, nám teda nestačí pozerať sa iba na samostatné neuróny (tak, ako sme to robili v kapitole 2 a 3), ale je nutné pozerať sa aj na to, ako sú neuróny prepojené a ako si odovzdávajú informácie medzi sebou. Inými slovami, skúmať neurónovú sieť.

4.1 Synapsie

Na to, aby sme sa mohli presunúť od modelovania jedného neurónu (alebo skupiny nezávislých neurónov) k modelovaniu neurónových sietí musíme vysvetliť, ako sú neuróny prepojené.

V podkapitole 1.2 sme si povedali, že sú prepojené cez synapsie. Dalej sme si povedali, že cez synapsie si vedia navzájom vysielať chemické impulzy.

Neurón, ktorý na danej synapsii impulz vysiela, sa zvykne nazývať presynaptický a neurón ktorý impulz dostáva $postsynaptický^{27}$.

Chemický impulz od presynaptického neurónu postsynaptickému neurónu je vyvolaný spikom v presynaptickom neuróne²⁸. Ako dôsledok rozšírenia spiku pozdĺž axónu sa uvoľnia chemikálie z terminálu axónu presynaptického neurónu do priestoru medzi neurónmi, tzv. synaptickej štrbiny (angl. *synaptic cleft*). Tieto chemikálie sa nazývajú neurotransmitery a sú v termináloch axónu uložené v tzv. synaptických vačkoch (angl. *synaptic vesicle*).

Na Obrázku 4.1 vľavo je schematická ilustrácia uvoľnenia neurotransmiterov z vačkov do synaptickej štrbiny, cez ktorú sa následne dostávajú k membráne dendritu postsynaptického neurónu a viažu sa na špeciálne receptory na tejto

 $^{^{27}}$ Termíny presynaptický a postsynaptický sa vždy viažu na jednu konkrétnu synapsiu medzi dvoma neurónmi.

²⁸ V práci budeme predpokladať, že sa neurotransmitery uvoľňujú len pri spiku. Avšak opačnú implikáciu nepredpokladáme, t.j. spike nemusí vždy spôsobiť uvoľnenie neurotransmiterov.

membráne. Vpravo je skutočná fotka z elektrónového mikroskopu. Obrázok 4.1 bol zobraný z [22].



Obrázok 4.1 – Uvoľnenie neurotransmiterov do synaptickej štrbiny

Synapsie delíme na excitačné a inhibičné. V prípade excitačných synapsií väzba neurotransmiterov na receptory na membráne postsynaptického neurónu spôsobí otvorenie chemicky riadených iónových kanálov v tejto membráne, ktoré spôsobí tok prúdu smerom do neurónu, a teda aj zvýšenie napätia na membráne. Ak napätie prekročí istú hranicu, tak sa otvoria aj iné iónové kanály, napätia sa zvýši ešte viac a postsynaptický neurón vygeneruje spike. Tento proces je podrobne opísaný v [34].

V prípade inhibičných synapsií tečie po príchode neurotransmiterov prúd cez chemicky riadené kanále smerom von z neurónu a napätie sa zníži. Toto zníženie môže kompenzovať zvýšenie napätia v dôsledku impulzov od iných neurónov a môže tak zabrániť prípadnému vygenerovaniu spiku.

V práci budeme často spomínať chemicky riadené kanály, pričom vždy budeme mať na mysli chemicky riadené kanály v membráne dendritov postsynaptického neurónu. Viac o stavbe terminálu axónu, synaptickej štrbiny a dendritu v [23],[15].

Medzi najčastejšie neurotransmitery patrí podľa [22] glutamát, kyselina gamma-aminobutyrická, dopamín a acetylcholín. Glutamát slúži u cicavcov najmä na excitáciu a kyselina gamma-aminobutyrická najmä na inhibíciu.

Celý proces uvoľnenia neurotransmiterov zaberá veľmi krátky čas a v práci budeme tento čas vo výpočtoch zanedbávať.

4.2 Chemicky riadené kanály

V predošlej podkapitole sme povedali, že príchod neurotransmiterov spôsobí otvorenie chemicky riadených iónových kanálov v membráne postsynaptického neurónu. V tejto podkapitole si vysvetlíme, ako tento proces prebieha. Nebudeme však podrobne vysvetľovať všetky biologické aspekty. Biologický pohľad na daný proces je rozpísaný napríklad v [15].

Budeme sa na problematiku pozerať najmä z matematického hľadiska. Avšak ani po matematickej stránke nebudeme všetky diferenciálne rovnice v nasledujúcich podkapitolách odvodzovať úplne podrobne. Je to z dôvodu, že najbližšie 2 podkapitoly netvoria jadro kapitoly 4, ale slúžia len na uvedenie čitateľa do problematiky modelov, ktoré budú predstavené v podkapitole 4.4. Tými sa už budeme venovať podrobnejšie a ukážeme aj viaceré počítačové simulácie.

Napätie na membráne neurónu sa dá popísať podľa nasledovného zjednodušeného²⁹ modelu:

$$C\frac{dV}{dt} = g(V_{\text{rest}} - V) + I_{\text{ext}}(t), \qquad (4.1)$$

kde V je aktuálne napätie, g je vodivosť membrány, V_{rest} je napätie membrány počas pokojového stavu (t.j. do neurónu neprichádza nijaký externý prúd), C je kapacita membrány a $I_{\text{ext}}(t)$ je externý prúd, ktorý do neurónu púšťame pomocou elektródy (používaný iba pri pokusoch pri rôznych simuláciách).

Pokiaľ zahrnieme aj chemicky riadené kanály, dostávame (podľa [3]) :

$$C \frac{dV}{dt} = g(V_{\text{rest}} - V) + I_{\text{ext}}(t) + G_s(t).(E_s - V).$$
 (4.2)

Člen $G_s(t)$ ja tzv. synaptická vodivosť, tá sa zvyšuje príchodom neurotransmiterov. Člen E_s je hranica, nad ktorú keď sa dostane napätie, tak už člen $G_s(t).(E_s - V)$ spôsobuje pokles napätia a nie nárast.

²⁹ Model je zjednodušený preto, že nezahŕňa tzv. napäťovo riadené kanály a ani chemicky riadené kanály. Je však v literatúre často používaný. Jeho podrobné vysvetlenie a odvodenie môžeme nájsť v [34]. Tu môžeme tak isto nájsť aj model, ktorý zahŕňa aj napäťovo riadené kanály (tzv. Hodgkin–Huxleyho model). Ten v tejto práci však pre zjednodušenie uvádzať nebudeme.

Pri excitačných synapsiách je člen E_s vyšší ako _{Vrest} a pri inhibičných nižší. To je logické, pretože ak je neurón v pokojovom stave (čiže $V(t_0) = V_{\text{rest}}$, $I_{\text{ext}}(t_0) = 0$, $G_s(t_0) = 0$), tak člen $(E_s - V)$ je pri excitačnej synapsii kladný a príchod neurotransmiterov (zvýšenie $G_s(t)$) spôsobí nárast napätia. Pri inhibičnej synapsii je to teda opačne.

4.2.1 Synaptická vodivosť

Synaptickú vodivosť $G_s(t)$ vieme napísať ako $G_s(t) = s(t).G_{\text{max}}$, kde G_{max} je maximálna možná synaptická vodivosť a s(t) je podiel otvorených chemicky riadených kanálov (postsynaptického) neurónu zo všetkých takých kanálov na danej synapsii³⁰ ($G_s(t)$ aj s(t) sa teda vždy viažu na jednu konkrétnu synapsiu). Ako už bolo spomenuté, chemicky riadené kanály sa otvárajú príchodom neurotransmiterov od presynaptického neurónu do synaptickej štrbiny a neurotransmitery sa uvoľňujú v dôsledku spiku v presynaptickom neuróne. Čím viac spikov teda presynaptický neurón generuje, tým vyššie je s(t) (a teda aj $G_s(t)$) postsynaptického neurónu. Vzhľadom na to, že $G_s(t)$ ovplyvňuje priamo napätie na membráne neurónu a napätie na membráne neurónu má (pri dostatočnom zvýšení) za následok vygenerovanie spiku, môžeme tvrdiť, že spiky presynaptického neurónu ovplyvňujú spiky postsynaptického neurónu, a to práve cez veličinu $G_s(t)$. Práve pre tento fakt, bude $G_s(t)$ (resp. jej súčasť s(t)) jedna z kľúčových premenných, na ktoré sa vo zvyšku kapitoly 4 zameriame.

Podľa [13] môžeme synaptickú vodivosť modelovať pomocou nasledovných vzťahov:

$$\frac{dG_s}{dt} = \frac{-G_s}{\tau_s} \tag{4.3}$$

a

$$G_s(t) \rightarrow G_s(t) + \Delta G \text{ ak } t \in \{t_1, t_2, \dots, t_n\},$$

$$(4.4)$$

kde $\{t_1, t_2, ..., t_n\}$ je množina časov spiku a τ_s je časová konštanta.

 $^{^{30}}$ Písmeno s budeme teda v tejto kapitole označovať inú premennú, ako v predošlých kapitolách.

Rovnica (4.3) hovorí o exponenciálnom poklese synaptickej vodivosti k nulovej hodnote pri neprítomnosti presynaptických spikov. Zodpovedá to postupnému zatváraniu chemicky riadených kanálov. Tie totižto po príchode neurotransmiterov ostávajú nejaký čas otvorené aj bez toho, aby prichádzali ďalšie neurotransmitery.

Rovnica (4.4) hovorí o skokovitom náraste³¹ synaptickej vodivosti pri príchode presynaptického spiku. Hodnota ΔG , o ktorú synaptická vodivosť narastie, je daná tým, ako jednotlivé presynaptické spiky ovplyvňujú otváranie chemicky riadených kanálov.

Definujme $P_{\rm rel}$ ako pravdepodobnosť, že synaptický vačok v presynaptickom neuróne pri spiku uvoľní neurotransmiter. Predpokladajme, že pre všetky vačky je táto pravdepodobnosť rovnaká. Vzhľadom na veľké množstvo vačkov na termináli axónu, môžeme pravdepodobnosť $P_{\rm rel}$ považovať za ekvivalentnú s podielom počtu vačkov, ktoré pri spiku uvoľnia neurotransmiter a ich maximálneho počtu³². Vo zvyšku práce budeme s oboma pojmami pracovať ako so synonymami.

Definujme a_0 ako pravdepodobnosť, že sa zatvorený postsynaptický chemicky riadený kanál po príchode neurotransmitera otvorí. Podobne ako v predošlom odseku, vzhľadom na veľký počet kanálov, budeme a_0 interpretovať aj ako spomínanú pravdepodobnosť a aj ako podiel zatvorených kanálov, ktoré sa otvoria pri vypustení maximálneho množstva neurotransmitera z ich celkového počtu.

Zmena premennej s(t) po príchode presynaptického spiku bude teda:

$$s(t) \rightarrow s(t) + P_{\rm rel}a_0(1 - s(t)). \tag{4.5}$$

Rovnica (4.5) v slovnom vyjadrení znamená, že podiel otvorených kanálov sa po presynaptickom spiku zvýši o tie, ktoré sa otvoria v dôsledku uvoľnenia

³¹ Takýto skokovitý nárast je najjednoduchší spôsob, ako modelovať nárast synaptickej vodivosti pri príchode spiku. V [13] a [5] sú uvedené aj iné príklady nárastu synaptickej vodivosti. Vo všeobecnosti sa však vždy jedná o veľmi rapídny nárast, a preto tento najjednoduchší model často úplne stačí.

³² V realite sa počet vačkov a teda množstvo neurotransmitera v termináloch axónov líši medzi jednotlivými časťami tela. Spojenia mozgu so svalmi musia obsahovať veľké množstvo neurotransmitera, pretože svalové reakcie musia byť spoľahlivé a bez náhodnosti. Synapsie v rámci mozgu, týkajúce sa komplexných myšlienok, majú niekedy neurotransmitera menšie množstvo a tam už nenáhodnosť, ktorú predpokladáme, platiť nemusí.

neurotransmitera. Otvoriť sa môžu iba zatvorené chemicky riadené kanály, ktorých podiel je (1-s(t)) z celkového počtu. Z nich (otvorených) sa pri príchode neurotransmitera otvorí podiel a_0 . Vzťah berie do úvahy aj skutočnosť, že $P_{\rm rel}$ je podiel synaptických vačkov, ktoré pri spiku uvoľnia neurotransmiter (z ich maximálneho počtu). Percento novootvorených kanálov bude teda $P_{\rm rel}a_0(1-s(t))*100$.

Vzhľadom na (4.5) a fakt, že $G_s(t)=s(t).G_{\max}$ dostávame $\Delta G=G_{\max}P_{\mathrm{rel}}a_0(1-\mathbf{s}(t)).$

Naša interpretácia faktoru (1-s(t)) nie je po biologickej stránke úplne presná. Dôvod je taký, že príchod neurotransmitera neotvára priamo chemicky riadené kanály, ale aktivuje receptory na dendritoch, ktoré následne kanály otvoria. Ak by sa receptory vždy po príchode neurotransmitera aktivovali, tak každý uvoľnený neurotransmiter by spôsoboval otvorenie kanálu, až kým by neboli otvorené všetky chemicky riadené kanály. Člen (1-s(t)) alebo resp. $a_0(1-s(t))$ by v tomto prípade nemalo zmysel do modelu zahŕňať a stačilo by s(t) ohraničiť zhora jednotkou. My však predpokladáme, že receptor sa po príchode neurotransmitera na nejaký čas "nasýti" a nie je schopný otvárať ďalšie chemicky riadené kanály. Toto "nasýtenie" sa nazýva synaptická saturácia a je bližšie popísaná v [19],[7]. Tak isto pre zjednodušenie predpokladáme, že podiel nasýtených receptorov je ekvivalentný s podielom otvorených kanálov (toto zjednodušenie sa využíva aj v niektorých modeloch popísaných v [13]). Teda zahrnutie faktoru (1-s(t)) do modelu má zmysel.

4.3 Krátkodobá synaptická plasticita

 $P_{\rm rel}$ sa môže v čase meniť a môže byť ovplyvnené aj samotnými spikmi. Spôsobuje to tzv. krátkodobá synaptická plasticita³³.

Krátkodobá synaptická plasticita je niečo ako krátkodobá "pamäť" synapsie. Prejavuje sa tak, že jednotlivé presynaptické spiky neurónu znižujú resp. zvyšujú

³³ Existuje aj dlhodobá synaptická plasticita. Tou sa však v práci nezaoberáme. Preto ak niekedy v práci spomenieme iba "plasticitu", budeme myslieť práve krátkodobú.

pravdepodobnosť uvoľnenia neurotransmiterov pri presynaptických spikoch po nich nasledujúcich.

Pokiaľ ide o zníženie, hovoríme o synaptickej depresii, a pokiaľ ide o zvýšenie, hovoríme o synaptickej facilitácii.

4.3.1 Synaptická depresia

Synaptická depresia je spôsobená tým, že v termináloch axónu je síce veľký, ale obmedzený počet vačkov obsahujúcich neurotransmiter. Tento počet sa bezprostredne po príchode spiku zníži (o tie, z ktorých sa neurotransmiter uvoľnil do synaptickej štrbiny). Preto ak by prišlo veľa spikov za sebou a vačky sa nestíhali dopĺňať, nemohol by sa vždy uvoľniť plný počet vačkov.

Podiel vačkov prítomných v termináli axónu a ich maximálneho počtu, budeme označovať ako D(t). Bude to teda číslo z intervalu <0,1>. Podľa [13] sa vačky dopĺňajú podľa rovnice:

$$\frac{dD}{dt} = \frac{1-D}{\tau_D} \tag{4.6}$$

a po príchode spiku sa D(t) zmení podľa vzťahu:

$$D(t) \rightarrow D(t) - P_{\rm rel}(t).$$
 (4.7)

Zmena (4.7) je logická, pretože hovorí o tom, že podiel vačkov s neurotransmiterom sa zníži o toľko, koľko z nich svoj obsah z neurónu uvoľní. τ_D je časová konštanta určujúca, ako rýchlo sa vačky dopĺňajú. $P_{\rm rel}(t)$ sa teda dá napísať ako $p_r.D(t)$, kde p_r je pravdepodobnosť, že vačok neurotransmiter uvoľní za podmienky, že je (vačok) pripravený v termináli (táto podmienka robí rozdiel medzi p_r a $P_{\rm rel}(t)$).

4.3.2 Synaptická facilitácia

Synaptická facilitácia je proces, pri ktorom spike spôsobí, že sa zvýši pravdepodobnosť p_r uvoľnenia neurotransmitera (za podmienky, že je v termináli). Tento proces je spôsobený viacerými biologickými faktormi, ktoré uvádzať nebudeme, ale dajú sa pozrieť v [9]. Synaptická facilitácia je reprezentovaná členom F(t). Ten hovorí o tom, ako veľmi zvýši spike pravdepodobnosť p_r . Pravdepodobnosť $p_r(t)$ budeme vyjadrovať ako $F(t).p_0$, kde p_0 je konštanta, ktorá hovorí o hodnote p_r po dlhom "tichu" (po dlhej absencii spikov).

Premenná F(t) bude teda nadobúdať hodnoty z intervalu $<0, 1/p_0>$. Horná hranica je daná tým, že $p_r(t)$ je pravdepodobnosť a nesmie teda prekročiť 1.

F(t) sa bude podľa [13] správať podľa rovnice

$$\frac{dF}{dt} = \frac{1-F}{\tau_F} \tag{4.8}$$

a po príchode spiku sa F(t) zmení podľa vzťahu:

$$F(t) \rightarrow F(t) + f_{\text{fac}}(1/p_0 - F(t)),$$
 (4.9)

kde $f_{\rm fac}$ je facilitačná konštanta z intervalu <0,1>, určujúca mieru facilitácie. Ak je facilitácia maximálna, čiže $f_{\rm fac} = 1$, tak (podľa (4.9)) stúpne F(t) pri každom spiku priamo na svoju maximálnu hodnotu $1/p_0$. Ak je facilitácia nulová, čiže $f_{\rm fac} = 0$, tak F(t) pri spiku nestúpne vôbec.

4.3.3 Dôsledky krátkodobej synaptickej plasticity

Diferenciálne rovnice (4.6) a (4.8) sú lineárne a separáciou premenných a následnou integráciou sa dajú vyriešiť. Ich vyriešením a použitím vzťahu $P_{\rm rel}(t)=p_0D(t)F(t)$ dostávame, že $P_{\rm rel}(t)$ sa počas obdobia bez spiku správa podľa rovnice:

$$P_{\rm rel}(t) = p_0 \left(1 - e^{-\frac{t}{\tau_D}} (1 - D(t_i)) \right) \left(- e^{-\frac{t}{\tau_F}} (1 - F(t_i)) \right), \tag{4.10}$$

kde t_i je čas posledného spiku a po príchode spiku sa zmení podľa:

$$P_{\rm rel}(t_{i+1}) \longrightarrow p_0(F(t_{i+1}) + f_{\rm fac}(1/p_0 - F(t_{i+1})))(D(t_{i+1}) - P_{\rm rel}(t_{i+1})).$$
(4.11)

Na (4.11) si môžeme všimnúť aj to, že pri spiku sa najprv zmení D a až potom F, čo je vzhľadom na to, ako bola zmena D konštruovaná intuitívne.

Je dôležité si uvedomiť, že mieru depresie určuje konštanta τ_D . Čím je táto konštanta nižšia, tým rýchlejšie sa vačky do terminálu dopĺňajú a tým menší efekt

depresia má. Pri veľmi nízkych hodnotách τ_D sa vačky stihnú doplniť všetky ešte pred najbližším možným príchodom ďalšieho spiku³⁴.

Intenzitu facilitácie zase určujú konštanty τ_F a f_{fac} . Druhá hovorí o veľkosti zmeny pri spiku a prvá o tom, ako dlho táto zmena vydrží (opäť platí, čím sú nižšie, tým je facilitácia slabšia).

Konštanty τ_D , τ_F , f_{fac} sa vždy viažu na presynaptický neurón a medzi jednotlivými synapsiami sa líšia, preto sa v literatúre často rozlišujú pojmy facilitačná synapsia a depresívna synapsia. Toto rozdelenie však nie je binárne a na facilitáciu a depresiu každej synapsie by sa skôr malo pozerať samostatne. Facilitačnými synapsiami budeme teda v práci nazývať tie, ktoré majú silnú facilitáciu a depresívnymi tie, ktoré majú silnú depresiu. Môže sa teda stať, že synapsia bude aj facilitačná aj depresívna.

V práci budeme vždy predpokladať, že neurón má pre všetky svoje synapsie, kde je presynaptický, rovnaké konštanty τ_D , τ_F , f_{fac} . To, či budú mať v modeli všetky neuróny tieto konštanty rovnaké, alebo či sa budú medzi neurónmi v modeli líšiť, bude závisieť od modelu a vždy to uvedieme.

Krátkodobá synaptická plasticita ovplyvňuje synaptický prenos informácie rôzne. Napríklad prenos signálu cez synapsie s vysokou depresiou je závislý najmä na prudkých nárastoch firing rate presynaptického neurónu a nie na nejakej jej stabilnej hodnote. Je to preto, že ak by aj firing rate bola vysoká (ale stabilná) a neurón by teda pálil spiky často, synaptická depresia by spôsobovala to, že jednotlivé spiky by neuvoľňovali veľa neurotransmiterov, a teda by ovplyvňovali napätie (a teda aj spiky) postsynaptického neurónu len veľmi málo. Pri takýchto neurónoch je teda vplyv na postsynaptický neurón pri osamotených spikoch a pri sériách s veľa spikami za sebou takmer rovnaký. Preto majú aj osamotené spiky na postsynaptický neurón relatívne vysoký vplyv.

³⁴ Neurón má nejakú refrakčnú periódu bezprostredne po vygenerovaní spiku, počas ktorej nie je schopný generovať ďalšie spiky, pozri [15].

Synaptická facilitácia zase spôsobuje to, že čím viac neurón páli, tým viac neurotransmiterov každým spikom uvoľňuje a tým viac ovplyvňuje napätie postsynaptického neurónu. Takéto neuróny preto najsilnejšie ovplyvňujú postsynaptický neurón pri sériách s veľa spikami za sebou. Treba ešte pripomenúť, že vyššia facilitácia je možná len pri synapsiách s nízkou p_0 , keďže v prípade vysokej p_0 je aj bez facilitácie pravdepodobnosť uvoľnenia neurotransmiterov vysoká a nemá už veľmi kam stúpnuť. Typický neurón s facilitačnými synapsiami teda pri osamotených spikoch veľmi postsynaptický neurón neovplyvní (vzhľadom na nízke p_0), ale ovplyvní ich silno už spomínanými sériami.

4.4 Firing rate model

Firing rate model je základný model používaný na popísanie fungovania neurónovej siete. Popisuje ako jednotlivé synaptické vstupy (angl. *synaptic inputs*), ktoré neurón dostáva od neurónov, na ktoré je napojený, ovplyvňujú jeho firing rate, a ako následne jeho firing rate ovplyvňuje to, aké synaptické vstupy odovzdá ostatným neurónom, ktoré sú naňho napojené.

Pri modelovaní neurónových sietí sa zvyknú jednotlivé neuróny agregovať do tzv. jednotiek. Do jednotky agregujeme určitý počet neurónov s podobnými vlastnosťami. Týmto sa výpočty značne zjednodušujú, keďže nemusíme uvažovať veľké množstvo neurónov a sledovať aktivitu každého samostatne, ale stačí sledovať priemernú aktivitu neurónov v danej jednotke. Spájanie neurónov do jednej jednotky spôsobuje, že jednotka býva často napojená aj sama na seba. Toto napojenie reprezentuje prepojenia neurónov v rámci jednotky.

Za firing rate jednotky budeme označovať priemernú firing rate všetkých neurónov, ktoré jednotka obsahuje³⁵.

 $^{^{35}}$ Počítame teda pre každý neurón firing rate z Definície 1 z druhej kapitoly. Takto vypočítané firing rate neurónov v rámci jednotky potom priemerujeme. Netreba si to zmýliť s priemernou firing rate zo vzťahu (2.3).

V najbližších 4 častiach budeme pre zjednodušenie takto vytvorené jednotky interpretovať (a často aj označovať) jednoducho ako neuróny. Budeme teda pracovať výhradne na úrovni jednotiek a nie na úrovni jednotlivých neurónov.

Dalším zjednodušením v modeli je skutočnosť, že neberieme do úvahy jednotlivé časy spikov, ale iba firing rate. Najväčšou výhodou tohto prístupu je to, že je jednoduchší a vďaka tomu máme možnosť lepšie pomôcť pochopiť správanie sa jednotlivých sietí. Tak isto majú takéto modely aj menej premenných (netreba si pamätať jednotlivé časy spikov) a tým sú výpočtovo oveľa rýchlejšie.

Nevýhodou firing rate modelov je, že strácame informáciu o možných koreláciach medzi spikmi a presným časovaním jednotlivých spikov, keď že firing rate neberie do úvahy presné časy spikov.

Definícia 2. Pod slovným spojením "neurón A je napojený na neurón B" budeme rozumieť, že existuje synapsia medzi neurónom A a neurónom B, na ktorej je neurón A postsynaptický a neurón B presynaptický.

Definícia 3. Pod slovným spojením "jednotka A je napojená na jednotku B" budeme rozumieť, že existuje aspoň jeden neurón v jednotke A, ktorý je napojený na nejaký neurón v jednotke B.³⁶

Definícia 4. Pre všetky $i \in I$ budeme symbolom J_i označovať množinu všetkých takých indexov $j \in I$, že neurón/jednotka s indexom i je napojený/á na neurón/jednotku s indexom j.

Platí, že $J_i \subset I.$

4.4.1 Synaptický vstup

Interpretácia synaptického vstupu sa v literatúre líši. V niektorých zdrojoch sa ako synaptický vstup uvažuje prúd do postsynaptického neurónu spôsobený otvorením chemicky riadených kanálov v postsynaptickom neuróne. V iných zase synaptická vodivosť membrány $G_s(t)$ postsynaptického neurónu a v ďalších podiel otvorených

³⁶ Keďže v tejto podkapitole myslíme pod pojmom "neurón" jednotku neurónov, tak pre "neuróny" v tejto kapitole platí skôr Definícia 3.

chemicky riadených kanálov s(t) v postsynaptickom neuróne. Spomínané veličiny sú síce rozdielne, ale všetky majú spoločné to, že pozitívne vplývajú na firing rate postsynaptického neurónu, a preto ostáva princíp výsledného firing rate modelu pri všetkých interpretáciách synaptického vstupu rovnaký. Zmenia sa iba voliteľné funkcie a konštanty, keďže synaptické vstupy sú pri rôznych interpretáciách v rôznych jednotkách. My budeme v práci používať poslednú spomenutú interpretáciu, to znamená, že synaptický vstup s(t) je podiel otvorených chemicky riadených kanálov postsynaptického neurónu.

Definícia 5. Na synapsii, kde je i-ty neurón presynaptický a j-ty neurón postsynaptický budeme podiel otvorených chemicky riadených kanálov postsynaptického neurónu označovať ako $s_i(t)$.³⁷

Premennú $s_i(t)$ budeme v práci označovať aj ako "vstup, ktorý *i*-ty neurón posiela/dáva/odovzdáva neurónom, ktoré sú naňho napojené".

i-ty neurón bude teda dostávať od neurónov, na ktoré je napojený, vstupy $s_i(t)$, pričom $j \in J_i$.

Celkový synaptický vstup, ktorý dostáva *i*-ty neurón, budeme označovať ako $s^i(t)^{38}$, pričom:

$$s^{i}(t) = rac{1}{\sum_{j \in J_{i}} (W_{ji})} \sum_{j \in J_{i}} W_{ji} s_{j}(t)$$
, (4.12)

kde W_{ji} je konštanta, nazývaná synaptická váha (angl. synaptic weight), ktorej veľkosť v absolútnej hodnote určuje silu spojenia medzi *j*-tym a *i*-tym neurónom. Celkový synaptický vstup je teda súčet jednotlivých vstupov od neurónov s indexom z J_i , prenásobených ich príslušnými synaptickými váhami. Navyše je predelený konštantou $1/\sum_{j \in J_i} (W_{ji})$, ktorá zaručuje, že aj $s^i(t)$ bude hodnota medzi 0 a 1.

Pokiaľ je na každej synapsii otvorený rovnaký podiel kanálov, napr. $\frac{1}{2}$, tak aj $s^{i}(t)$ nadobudne hodnotu $\frac{1}{2}$ a bude to znamenať, že je otvorená presne polovica

³⁷ Spodný index "*i*" na synaptickom vstupe je v takomto kontexte relatívne neintuitívny a možno by sa mohlo zdať, že tam má byť "*j*". My však (tak, ako v [13]) uvažujeme model, v ktorom sa síce synaptický vstup týka postsynaptického neurónu, ale závisí výhradne od presynaptického. V časti 4.4.3 vysvetlíme podrobnejšie, prečo je tento spôsob uvažovania korektný.

³⁸ Všetky vzorce a úvahy o s(t), odvodené v podkapitolách 4.2 a 4.3 platia len pre $s_i(t)$, nie pre $s^i(t)$.

zo všetkých chemicky riadených kanálov daného neurónu. Pokiaľ podiel otvorených kanálov $s_j(t)$ nebude na každej synapsii rovnaký, $s^i(t)$ už nebude vyjadrovať presný celkový podiel všetkých otvorených kanálov neurónu, ale vážený podiel, pri ktorom je zohľadnená aj sila jednotlivých synapsií.

4.4.2 Vstupno-výstupná funkcia

Kľúčovým prvkom firing rate modelu je tzv. vstupno-výstupná funkcia (angl. *input*output function). Vstupno-výstupná funkcia hovorí o závislosti firing rate *i*-teho neurónu (jednotky) $r_i(t)$ od jeho celkového vstupu $s^i(t)$, ktorý dostáva od neurónov, na ktoré je napojený. Túto funkciu budeme pre *i*-ty neurón označovať ako f_i .³⁹ Funkcia f sa zvykne podľa [22] uvažovať ako jeden z nasledovných typov (prípadne ich násobkov alebo iných modifikácií):

$$egin{aligned} f(x) &= x, \ f(x) &= \lfloor (1 + \operatorname{sign}(x)
floor, \ f(x) &= x + \lfloor x
floor, \ f(x) &= (1 + e^{-x})^{-1}, \ f(x) &= x - \lfloor x - 1
floor + 1. \end{aligned}$$

Funkcia f by mala na svojom definičnom obore spĺňať nasledovné vlastnosti:

- 1. neklesajúcosť,
- 2. kladnosť,
- 3. ohraničenosť.

Neklesajúcosť je daná tým, že čím vyššie synaptické vstupy $s_j(t)$ (viac otvorených kanálov, a teda aj viac prúdu tečúceho do neurónu), tým vyššia je šanca, že napätie na neurálnej membráne prekročí prahovú hranicu a neurón vygeneruje spike.

Vzhľadom na neklesajúcosť vstupno-výstupnej funkcie, musia byť váhy W pre inhibičné synapsie záporné. Je to spôsobené tým, že na inhibičných synapsiách príchod neurotransmiterov otvára chemicky riadené kanály, cez ktoré tečie prúd von

 $^{^{39}}$ Písmenom "f" sme už v práci niektoré funkcie ako f označili. Táto funkcia s nimi nijako nesúvisí a toto označenie volíme vzhľadom na jeho veľkú rozšírenosť v literatúre v tejto súvislosti.

z neurónu. Ak je teda synapsia medzi neurónmi j a i inhibičná, otvorenie kanálov (zvýšenie $s_j(t)$) má na $s^i(t)$ negatívny vplyv. V prípade neurónu, ktorý má aj inhibičné (a teda aj záporné W) aj excitačné vstupy sa celkový synaptický z (4.12) musí počítať samostatne pre inhibičné a samostatne pre excitačné synapsie. Následne sa celkový vstup získa odčítaním takto vypočítaného celkového excitačného a celkového inhibičného vstupu.

V realite (vo vzťahu (4.2)) je znamienko synaptickej váhy reprezentované znamienkom člena ($E_s - V$).

Kladnosť vstupno-výstupnej funkcie je daná tým, že firing rate, ako výstup vstupno-výstupnej funkcie *f*, musí byť kladná (pozri kap. 2).

Ohraničenosť je daná tým, že každý neurón má obmedzenú maximálnu intenzitu generovania spikov.

f nemusí prechádzať bodom (0,0). To znamená, že aj pre nulové vstupy môže mať neurón nenulovú firing rate. Je to spôsobené tým, že neurón občas "vypáli" náhodný spike aj bez nejakej stimulácie. Takéto spiky berieme ako "šum".

Vstupno-výstupná funkcia je veľmi podobná statickej nelinearite z časti 2.3.3. Statická nelinearita hovorí o tom, ako vyzerá firing rate neurónu vzhľadom na lineárnu filtráciu stimulu. Lineárna filtrácia bola tým vyššia, čím podobnejší bol stimul s preferovaným stimulom neurónu. Pri vstupno-výstupnej funkcii sa nepozeráme na stimul, ale na vstup od neurónov, na ktoré je *i*-ty neurón napojený. To je však ekvivalentné, pretože preferencie neurónu sú dané práve tým, aké preferencie majú neuróny, na ktoré je neurón napojený (skladanie informácie z podkapitoly 1.3). Ak je teda prítomný preferovaný stimul pre *i*-ty neurón, tak nutne budú mať neuróny, na ktoré je neurón napojený vysokú firing rate (a teda aj vysoké synaptické vstupy pre *i*-ty neurón) a zároveň, ak posielajú neuróny *i*-temu neurónu vysoké vstupy, určite je prítomný preferovaný stimul *i*-teho neurónu. Jediný rozdiel je teda v tom, že pri LNP modeli sa pozeráme na úplne prvý vstup "informačnej dráhy" (ktorej príklad je v časti 1.3.2), ktorým je stimul. Pri firing rate modeli sa pozeráme vždy na predposledný vstup pre *i*-ty neurón, ktorým sú synaptické vstupy od neurónov, na ktoré je *i*-ty neurón napojený.
4.4.3 Rovnice popisujúce firing rate model

Firing rate model má v literatúre ([22],[17],[5],[13]) všeobecný princíp, pri ktorom závisí firing rate neurónu i od synaptických vstupov od všetkých neurónov, na ktoré je napojený a od nej (firing rate) následne závisí synaptický vstup, ktorý dáva i-ty ďalším neurónom, ktoré sú napojené naňho.

Tento princíp sa dá podľa [13] opísať systémom nasledovných diferenciálnych rovníc:

$$\tau_{r_i} \frac{dr_i}{dt} = -r_i + f_i \left(\sum_{j \in J_i} W_{ji} s_j(t) \right)$$
(4.14)

a

$$\tau_{s_i} \frac{ds_i}{dt} = -s_i + F_i(r_i(t)).$$
(4.15)

 W_{ji} je synaptická váha medzi *j*-tym a *i*-tym neurónom, pričom *j*-ty neurón je v tejto synapsii presynaptický a *i*-ty postsynaptický.

Rovnica (4.15) vyplýva z rovnice (4.3). Pokiaľ do (4.3) dosadíme za $G_s(t) = s(t).G_{\text{max}}$, dostaneme

$$\tau_{s_i} \frac{ds_i}{dt} = -s_i, \tag{4.16}$$

čo vyjadruje zatváranie sa chemicky riadených kanálov. V (4.15) je okrem tohto zatvárania aj otváranie spôsobené príchodom spikov (ktoré je riadené firing rate). Toto otváranie je dané funkciou F.

 F_i je tzv. bránová funkcia (angl. gating function) *i*-teho neurónu. Bránová funkcia hovorí o tom, ako závisí s(t) od intenzity generovania spikov. Jej tvar závisí od toho, či do modelu zahŕňame alebo nezahŕňame synaptickú saturáciu, depresiu a facilitáciu. Pokiaľ nič z toho nezahŕňame, tak podľa [13] volíme bránovú funkciu ako lineárnu. Vzhľadom na to, že s(t) nadobúda hodnoty iba medzi 0 a 1, bude lineárna F vyzerať ako:

$$F_i(r_i) = \frac{r_i}{r_{\text{max}}} \,\forall i. \tag{4.17}$$

O tom, ako vyzerá F, ak saturáciu, depresiu alebo facilitáciu do modelu zahŕňame, si povieme neskôr.

Môže sa stať, že pre dané *i* a *j* existuje aj spojenie *ij* aj *ji* (t.j. napr. pre *j*=2 a *i*=3 môže platiť, že $3\in J_2$ a zároveň $2\in J_3$, čiže druhý neurón je napojený na tretí a tretí na druhý – tvoria niečo ako "kruh"). Taktiež môže nastať, že $i \in J_i$ t.j. neurón je napojený sám na seba⁴⁰.

 r_i označuje firing rate *i*-teho neurónu⁴¹.

 $s_i(t)$ je synaptický vstup, ktorý *i*-ty neurón odovzdáva (pozri Definícia 5) všetkým neurónom, ktoré sú naňho napojené. Pri interpretácii *s*, akú v práci používame, je teda $s_i(t)$ podiel otvorených chemicky riadených kanálov neurónov napojených na *i*-ty neurón. Mohlo by sa teda zdať, že pre *i*-ty neurón sa môže tento podiel ($s_i(t)$) v rámci neurónov, ktoré sú naňho napojené, líšiť, a preto by sme mali definovať skôr niečo ako $s_{ij}(t)$ pre $j \in J_i$. Tým by sa náš model značne skomplikoval. Je však dôležité si uvedomiť, že to, koľko chemicky riadených kanálov má postsynaptický neurón otvorených, je dané z prevažnej miery presynaptickým neurónom (ktorý je v našom prípade *i*) cez premennú $P_{\rm rel}$, a nie samotným postsynaptickým (v našom prípade neurón s indexom z množiny J_i). Postsynaptický neurón ovplyvňuje podiel otvorených kanálov len cez konštantu a_0 (pozri str. 63) , ktorá sa väčšinou uvažuje jednotná pre všetky neuróny. Aj keby sme a_0 uvažovali ako rozličnú medzi neurónmi, tak vzhľadom na to, že je to konštanta, ju vieme zahrnúť do konštanty W_{ij} , a preto bude princíp jednotného $s_i(t)$ pre *i*-ty neurón validný.

 f_i je vstupno-výstupná funkcia *i*-teho neurónu.

4.4.4 Rôzne interpretácie synaptického vstupu vo firing rate modeloch

To, že synaptický vstup $s_i(t)$ môže reprezentovať rôzne veličiny, sme už spomenuli v úvode tejto podkapitoly. Tak isto sme povedali, že sa na synaptický vstup i-teho

 $^{^{40}}$ Na pripomenutie, pod neurónom označujeme jednotku – agregát neurónov. Na str. 68,69 je o tom napísané viac. Tak isto aj o napojení jednotiek samých na seba.

 $^{^{41}}$ Opäť pod neurónom myslíme jednotku neurónov, a ted
a r_i označuje priemernú firing rate neurónov v jednotke.

neurónu môžeme pozerať rôznymi spôsobmi. Buď ako na $s_i(t)$, čiže ako na podiel otvorených chemicky riadených kanálov na synapsiách s *i*-tym neurónom tých neurónov, ktoré sú na *i*-ty neurón napojené – inými slovami, vstup, ktorý dáva *i*-ty neurón všetkým neurónom, ktoré sú naňho napojené. Alebo ako na $s^i(t)$, čiže ako na celkový synaptický vstup, ktorý *i*-ty neurón dostáva (vypočítateľný podľa (4.12)).

V literatúre (napr. v [5]) sa niekedy firing rate model vyjadruje nie pomocou interpretácie synaptického vstupu ako $s_i(t)$ ale ako $s^i(t)$:⁴²

$$\tau_{s^i} \frac{dI_s^i}{dt} = -I_s^i + \sum_{j \in J_i} \mathcal{W}_{ji} F_j(r_j(t)), \qquad (4.18)$$

$$au_{r_i} rac{dr_i}{dt} = -r_i + f_i(I_s^{\,i}(t)). ag{4.19}$$

Pričom funkciu F_i uvažuje ako lineárnu $F_j(x) = x \forall j$. V [5] sa navyše synaptický vstup uvažuje ako prúd $I_s(t)$ prúdiaci do neurónu. Definujeme teda konštantu I_{\max} takú, že: $I_s^i(t) = s^i(t)$. I_{\max} ktorá hovorí o tom, aký veľký prúd by tiekol do neurónu, ak by boli otvorené všetky jeho chemicky riadené kanály. Sústavu (4.18),(4.19) teda vieme prepísať ("do našej reči") na:

$$\tau_{s^{i}}\frac{ds^{i}}{dt} = -s^{i} + \frac{1}{I_{\max}}\sum_{j \in J_{i}} \mathcal{W}_{ji}^{*}r_{j}(t), \qquad (4.20)$$

$$\tau_{r_i} \frac{dr_i}{dt} = -r_i + f_i(s^i(t).I_{\max}). \tag{4.21}$$

Teraz ukážeme, že bez ohľadu na to, či berieme do úvahy $s_i(t)$ alebo $s^i(t)$, je firing rate model jednoznačný a sústavy (4.14),(4.15),(4.17) a (4.20),(4.21) sú ekvivalentné. Inak povedané, pre N neurónov, pre ľubovoľnú počiatočnú podmienku $r(0)=(r_1(0),r_2(0),...,r_i(0),...,r_N(0))$ a pre ľubovoľné množiny $J_1, J_2, ..., J_N$ sa bude vektor r(t) spĺňajúci rovnice (4.18) a (4.19) správať rovnako ako vektor r(t) spĺňajúci rovnice (4.14),(4.15),(4.17).

Vieme, že platí (4.12):

$$rac{1}{\sum_{j \in J_i}(W_{ji})} \sum_{j \in J_i} W_{ji} s_j(t) \ = \ s^i(t).$$

 $^{^{42}}$ V [5] sa používa iné značenie premenných ako používame my. Model, ktorý je tam uvedený na str. 235 sme v našej práci teda preznačili do nášho značenia.

Navyše budeme predpokladať, že konštanty τ_{s^i} a τ_{s_j} sa rovnajú, pretože obe hovoria o rýchlosti zatvárania sa chemicky riadených kanálov toho istého neurónu⁴³.

Definujme
$$\widetilde{W}_{ji} = \frac{1}{\sum_{j \in J_i} (W_{ji})} W_{ji} I_{\max}$$
, potom platí:

$$\sum_{j \in J_i} \widetilde{W}_{ji} s_j(t) = s^i(t) I_{\max}$$
(4.22)

a teda aj

$$\sum_{j \in J_i} \widetilde{W}_{ji} \frac{ds_j}{dt} = \frac{ds^i}{dt} I_{\max}$$
(4.23)

a teda pre r(t) spĺňajúce rovnice (4.14), (4.15), (4.16) platí

$$\frac{ds^{i}}{dt} = \sum_{j \in J_{i}} \frac{\widetilde{W}_{ji}}{I_{\max}} \frac{ds_{j}}{dt} = \sum_{j \in J_{i}} \frac{\widetilde{W}_{ji}}{I_{\max}\tau_{s_{j}}} \left(-s_{j} + \frac{r_{j}(t)}{r_{\max}} \right) = -\sum_{j \in J_{i}} \frac{\widetilde{W}_{ji}}{I_{\max}\tau_{s_{j}}} s_{j} + \sum_{j \in J_{i}} \frac{\widetilde{W}_{ji}}{I_{\max}\tau_{s_{j}}} \frac{r_{j}(t)}{r_{\max}}.$$
 (4.24)

Ak položíme

$$\mathcal{W}_{ji} = \frac{1}{\sum_{j \in J_i} (W_{ji})} W_{ji} \cdot \frac{I_{\max}}{r_{\max}}, \text{ čiže } \widetilde{W}_{ji} = W_{ji} \cdot r_{\max}$$
(4.25)

a využijeme rovnosť τ_{s^i} a τ_{s_i} , tak spojením (4.25),(4.24)(4.12) dostaneme (4.20).

Týmto sme ukázali dve veci. Prvá je, že to, či sa na *i*-ty synaptický vstup pozeráme ako na celkový vstup do *i*-teho neurónu alebo na vstup od *i*-teho neurónu, nemá vplyv na vývoj firing rate a nezmení sa ani štruktúra modelu a ani jeho komponenty ako vstupno-výstupná funkcia, bránová funkcia alebo diferenciálna rovnica pre r_i . Jediné, čo sa zmení, je vyjadrenie synaptického vstupu a jeho príslušná diferenciálna rovnica.

Navyše sme ukázali, že počítanie synaptickej váhy v inej veličine model (až na zmenu matice váh a oboru hodnôt bránovej funkcie) tiež neovplyvní, pokiaľ sa dajú veličiny vyjadriť ako násobok jedna druhej.

Na záver dodávame, že zmyslom časti 4.4.4 nebolo dokázať alebo objaviť nové spojitosti, prípadne existujúce spochybňovať. Časť 4.4.4 slúži skôr na to, aby si čitateľ čítajúci rôzne zdroje o firing rate modeloch uvedomil, že princíp ostáva

 $^{^{43}}$ Rovnosť τ_{s^i} a τ_{s_i} nepredpokladáme, keďže tie sa už viažu na chemicky riadené kanály rozdielnych neurónov.

rovnaký, aj napriek tomu, že sa v nich model (pričom najmä jeho synaptické vstupy) často prezentuje úplne inak. Naviac sa dá medzi jednotlivými interpretáciami "ľahko" prechádzať.

4.4.5 Synaptická saturácia a krátkodobá synaptická plasticita vo firing rate modeloch

V podkapitole 4.3 sme povedali o krátkodobej synaptickej plasticite. Teraz ukážeme, ako ich zakomponovať do firing rate modelu.

V časti 4.4.3 sme ukázali, že prvý člen pravej strany vzťahu (4.15) prislúchajúci postupnému zatváraniu sa chemicky riadených kanálov je odvodený zo vzťahu (4.3). Tak isto sme povedali, že bránová funkcia F prislúcha otváraniu sa chemicky riadených kanálov pri príchode spikov. Nakoľko vo vzťahu (4.15) nie je zahrnutá ani krátkodobá synaptická plasticita ani synaptická saturácia, mohli sme funkciu F zvoliť ako lineárnu. Ak však tieto vlastnosti synapsií zahrnieme, F nemôže byť lineárna, pretože pri krátkodobej synaptickej plasticite alebo saturácii neprináša každý spike rovnaké zvýšenie podielu otvorených chemicky riadených kanálov.

V časti 4.2.1 o zmene synaptickej vodivosti G_s (a teda aj o zmene podielu otvorených chemicky riadených kanálov s), pri zahrnutí synaptickej saturácie, hovoril vzťah (4.4) resp. (4.5). Navyše, ak vo vzťahu (4.5) neuvažujeme premennú $P_{\rm rel}(t)$ ako konštantnú, ale uvažujeme ju ako $P_{\rm rel}(t)=p_0D(t)F(t)$, dostávame model, ktorý zahŕňa aj krátkodobú synaptickú plasticitu.

Ak by sme sa otváranie chemicky riadených kanálov, ovplyvnené synaptickou saturáciou, rozhodli zahrnúť do firing rate modelu nie pomocou bránovej funkcie F, ale pomocou "skoku" v podiele otvorených kanálov pri príchode spiku (tak ako sme to robili v podkapitole 4.2.1 vo vzťahu (4.5)), dostali by sme model, tvorený vzťahmi (4.14),(4.16),(4.5):

$$egin{aligned} & au_{r_i} rac{dr_i}{dt} = -r_i + f_i \Biggl(\sum_{j \ \in J_i} W_{ji} s_j(t) \Biggr), \ & au_{s_i} rac{ds_i}{dt} = -s_i, \end{aligned}$$

$$s_i(t_k) \rightarrow s_i(t_k) + P_{\mathrm{rel}}a_0(1 - s_i(t_k)),$$

kde t_k je čas spiku.

Tento model má však jednu zásadnú chybu. Firing rate modely sú založené na firing rate a nie na jednotlivých spikoch. "Skokové" zvýšenie podielu otvorených chemicky riadených kanálov v čase spiku teda nevieme uskutočniť, keď že informáciu o časoch spikov nemáme. Otváranie kanálov pri príchode spiku, dané vzťahom (4.5), bude teda predsa len nutné zakomponovať do bránovej funkcie F. Teraz ukážeme ako.

Prvý krok je, že do diferenciálnej rovnice $\frac{ds_i}{dt} = -\frac{s_i}{\tau_{s_i}}$ sa dajú skoky premennej s_i z hodnoty $s_i(t_k)$ na hodnotu $s_i(t_k) + P_{rel}(t_k)a_0(1 - s_i(t_k))$ v časoch spiku t_k zakomponovať tak, že na jej pravú stranu pripočítame spike train, vyjadrený pomocou súčtu Diracových delta funkcií (funkcia neurálnej odpovede zo vzťahu (2.1)) prenásobený prírastkom, o ktorý s_i v čase spiku "poskočí":

$$\frac{ds_i}{dt} = -\frac{s_i}{\tau_{s_i}} + P_{\rm rel}a_0(1-s_i)\sum_{k=1}^n \delta(t-t_k).$$
(4.26)

Ekvivalencia rovnice (4.26) s rovnicami (4.16) a (4.5) sa dá ukázať vyriešením rovnice (4.26) pomocou *Laplacovej transformácie* popísanej v [25].

V zdrojoch, z ktorých čerpá táto práca, sa tiež do tejto diferenciálnej zakomponováva funkcia neurálnej odpovede. Spôsob odvodenia je však často menej priamy a menej plynulý.

V druhom kroku nahradíme funkciu neurálnej odpovede funkciou firing rate $r_i(t)$, vypočítanou pomocou vzťahu (2.5). Správnosť tohto nahradenia je popísaná v [5] a odvoláva sa na to, že takto vypočítaná firing rate je niečo ako hladká aproximácia funkcie neurálnej odpovede a na fakt, že v jednotke (ešte raz pripomíname, že to, čo v celej podkapitole 4.4 nazývame "neurón", je v skutočnosti jednotka pozostávajúca z viacerých neurónov), ktorá obsahuje veľký počet neurónov tak či tak priemerujeme spike trainy (funkcie neurálnej odpovede) jednotlivých neurónov. Tak isto je vhodné dodať, že validita tohto nahradenia je daná aj tým, že

funkcia neurálnej odpovede a firing rate, vypočítaná pomocou vzťahu (2.5), má na intervale <0, T> rovnaký určitý integrál.

Dostávame teda systém diferenciálnych rovníc:

$$\tau_{r_i} \frac{dr_i}{dt} = -r_i + f_i \left(\sum_{j \in J_i} W_{ji} s_j(t) \right)$$

$$\tau_{s_i} \frac{ds_i}{dt} = -s_i + \tau_{s_i} P_{\text{rel}} a_0 (1 - s_i) r_i(t).$$
(4.27)

a

Pri prevedení (4.27) na tvar (4.15) má bránová funkcia F tvar:

$$F_i(r_i(t)) = \frac{\tau_{s_i} P_{\text{rel}} a_0 r_i(t)}{1 + \tau_{s_i} P_{\text{rel}} a_0 r_i(t)} \forall i$$
(4.28)

a efektívna časová konštanta (konštanta pri derivácii, pokiaľ je člen $-s_i$,,osamotený" (v rovnici (4.15) je to τ_{s_i})) má tvar $\tau_{s_i}/(1+\tau_{s_i}P_{\text{rel}}a_0r_i)$.

Pokiaľ do firing rate modelu zahrnieme aj synaptickú plasticitu a vzťahy (4.6), (4.7), (4.8), (4.9) transformujeme rovnakým spôsobom, ako sme transformovali vzťahy (4.14), (4.5) na vzťah (4.27) (čiže spôsobom, pri ktorom najprv zahrnieme "skoky" zo samostatnej rovnice do diferenciálnej rovnice pridaním funkcie neurálnej odpovede, prenásobenej hodnotou "skoku", a následne funkciu neurálnej odpovede nahradíme funkciou firing rate, vypočítanou podľa vzťahu (2.5)), tak dostaneme nasledovný systém diferenciálnych rovníc (s $P_{rel}(t) = p_0 D(t)F(t)$):

$$\begin{split} \tau_{r_i} \frac{dr_i}{dt} &= -r_i + f_i \left(\sum_{j \in J_i} W_{ji} s_j(t) \right), \\ \frac{\tau_{s_i}}{1 + \tau_{s_i} (p_0 DF) a_0 r_i(t)} \frac{ds_i}{dt} &= -s_i + \frac{\tau_{s_i} (p_0 DF) a_0 r_i(t)}{1 + \tau_{s_i} (p_0 DF) a_0 r_i(t)}, \\ \frac{dD}{dt} &= \frac{1 - D}{\tau_D} - (p_0 DF) r_i(t), \\ \frac{dF}{dt} &= \frac{1 - F}{\tau_F} + f_{\text{fac}}.(1/p_0 - F) r_i(t). \end{split}$$

4.5 Simulácie firing rate modelu pri jednej jednotke s rekurentným excitačným spojením

Rekurentné napojenia sú napojenia jednotky samej na seba. Vďaka excitačným rekurentným konekciám sa môže stať, že jednotka sama seba excituje, a tak udržiava stabilnú firing rate aj pri nulových vonkajších vstupoch. Spomínané udržiavanie si konštantnej firing rate vďaka rekurentným konekciám sa nazýva stabilita.

Teraz si ukážeme rôzne simulácie takejto stability. Simulácie budú inšpirované cvičením "6.4" z knihy [13]. Budú však obohatené o rôzne nadstavby a špeciálne prípady, ktoré rozanalyzujeme.

Najprv budeme uvažovať len jednu jednotku s rekurentným napojením. Takéto spojenie je znázornené na Obrázku 4.2.



Obrázok 4.2 – Diagram jednej jednotky napojenej samú na seba.

Indexy i a j v tom
to prípade nie sú potrebné, nakoľko máme len jednu jednotku.

Na diagrame vidíme to, o čom sme hovorili v podkapitole 4.4, že jednotka (ktorú sme v častiach 4.4.2, 4.4.3, 4.4.4, 4.4.5 označovali zjednodušene ako "neurón") pozostáva z viacerých neurónov, z ktorých každý ma svoju firing rate. Tieto firing rate sa následne spriemerujú do výslednej firing rate jednotky r. Následne sa táto firing rate pomocou vstupno-výstupnej funkcie transformuje do synaptického vstupu s. Tento vstup sa v príslušných jednotkách (v tomto prípade je len jedna a keďže máme rekurentné spojenie, je to samotná jednotka, ktorá vstup vysiela) prenásobí synaptickou váhou W zahŕňajúcou silu spojenia, jeho excitačnosť/inhibičnosť

a v niektorých modeloch aj konštantu a_{0} , spomínanú v časti 4.2.1. Pod silou spojenia sa myslí napr. to, koľko neurónov z odosielajúcej jednotky je napojených na neuróny z prijímajúcej jednotky.

4.5.1 Model bez synaptickej saturácie, synaptickej plasticity a bez externého vstupu

Model popisujúci takúto "siet" bez zahrnutie saturácie a plasticity bude tvorený nasledovnými rovnicami:

$$egin{aligned} & au_r rac{dr}{dt} = -r + f\left(W.s(t)
ight), \ & au_s rac{ds}{dt} = -s + F(r(t)). \end{aligned}$$

Vstupno-výstupnú funkciu budeme voliť ako: $f(x) = \frac{100}{(1+e^{-10x+3})}$ a bránovú funkciu tak, ako v (4.16), pričom $r_{\max} = \max(f(x)) = 100$. Časové konštanty τ_s a τ_r a synaptickú váhu W budeme v jednotlivých simuláciách tohto modelu meniť.

Obrázky 4.3, 4.5, 4.6, 4.7, 4.10, 4.11, 4.12, 4.13 prislúchajú rôznym modelom, ktoré budeme v tejto podkapitole postupne predstavovať. Každý riadok týchto obrázkov prislúcha jednej simulácii. V každej simulácii sú použité rôzne parametre daného modelu⁴⁴, aby sme ukázali ich vplyv na vývoj premenných r a s.

Obrázok 4.3. zodpovedá vyššie predstavenému modelu bez synaptickej saturácie a plasticity. V jeho ľavej časti sú znázornené statické stavy jednotky. Oranžová krivka vyjadruje závislosť premennej s od r, pokiaľ je derivácia podľa s nulová. Modrá krivka vyjadruje závislosť premennej r od s, pokiaľ je derivácia podľa r nulová. Obe krivky teda vyjadrujú hodnotu druhej premennej, pokiaľ by bola tá prvá konštantná. V našom modeli je teda statický synaptický vstup rovný F(r) a statická firing rate rovná f(W.s).

⁴⁴ Parametre sa v skutočnosti viažu na konkrétnu jednotku. Korektné by teda bolo namiesto "parametre simulácie" alebo "parametre modelu" hovoriť "parametre jednotky, ktorú simulujeme" alebo "parametre jednotky, ktorú popisujeme daným modelom".

Rozdiel medzi modrou funkciou r(s) a oranžovou funkciou s(r) je v tom, že r(s) dostáva vstupy z horizontálnej osi s a dáva výstupy na vertikálnu os r (čiže ako bežné vykreslenie funkcie) a s(r) má vstupy z vertikálnej osi r a výstupy na horizontálnu os s. Funkcia s(r) je teda akoby nakreslená "obrátene", čiže jej definičný obor je vertikálna os a obor hodnôt horizontálna.

Treba si uvedomiť, že v našom prípade je jednotka napojená sama na seba a nemá žiadne iné spojenia, preto jej firing rate je závislá iba od jej vlastného



Obrázok 4.3 – Simulácie firing rate modelu s jednou jednotkou a bez zahrnutia synaptickej saturácie a synaptickej plasticity a bez externého vstupu.

synaptického vstupu.

Body, v ktorých sa modrá a oranžová krivka pretínajú, sú tzv. pevné body (angl. *fixed point*) systému, v ktorých sú obe derivácie nulové a firing rate ani synaptický vstup sa nemenia. Pevné body vieme okrem matematického výpočtu cez stacionárne riešenie nájsť aj tak, že pre počiatočnú hodnotu synaptického vstupu (bod na oranžovej krivke) urobíme zvislú projekciu na modrú krivku, ktorá vyjadruje firing rate jednotky pri takejto hodnote synaptického vstupu. Následne pre túto hodnotu firing rate (bod na modrej krivke) urobíme vodorovnú projekciu na oranžovú krivku a dostaneme tak príslušnú hodnotu synaptického vstupu pre danú firing rate. Tento proces vieme iteračne opakovať dovtedy, kým prídeme do pevného bodu. Takýto postup je graficky znázornený na Obrázku 4.4.



Obrázok 4.4 – Konvergencia do stabilného pevného bodu pri rôznych štartovacích bodoch

Pevný bod systému môže byť stabilný alebo nestabilný. Pri stabilnom bode má systém pri malom vychýlení sa z pevného bodu (napr. ako dôsledok externého vstupu) tendenciu vrátiť sa doňho. Pri nestabilnom pevnom bode je systém stabilný iba priamo v danom bode, ale pri hocijakom vychýlení sa z neho majú hodnoty tendenciu vzďaľovať sa od neho. Napríklad v druhej simulácii máme 3 pevné body, z ktorých dva sú stabilné a jeden (stredný) je nestabilný. To sme mohli vidieť aj na Obrázku 4.4 (ktorý zodpovedá práve druhej simulácii), kde sa spomínaným postupom s projekciami nikdy nedostaneme do stredného bodu, ale vždy buď do horného alebo do dolného, v závislosti od toho, aké štartovacie *s* si vyberieme. V pravej časti Obrázku 4.3 je simulovaný priebeh firing rate v čase po dobu jednej sekundy pri počiatočnej podmienke s nulovým synaptickým vstupom r(t=0)=f(s=0). Je vidno, že firing rate po určitom čase vždy dosiahne svoj najbližší stabilný pevný bod a ďalej sa nemení.

V prvých troch simuláciách bolo W postupne volené ako 0.2, 0.6, 1.2 a v posledných troch ako 5. Vidíme, že čím je synaptická váha vyššia, tým je samoexcitácia jednotky silnejšia a jednotka je tak schopná udržiavať vyššie hodnoty stabilnej firing rate.

Casové konštanty boli v prvých 3 pokusoch $\tau_s = 0.002$ a $\tau_r = 0.01$. V štvrtom pokuse sme zmenili τ_r na 0.1 a v piatom τ_s na 0.02 (pričom τ_r sme vrátili na 0.01). Je vidno, že čím vyššia je časová konštanta, tým dlhšie trvá jednotke dosiahnuť svoju stabilnú hodnotu firing rate. Vidíme aj to, že systém je v tomto citlivejší na premennú τ_r ako na τ_s , keďže zmena τ_s na 0.02 (pri šiestej simulácii) spôsobila len minimálne spomalenie nárastu firing rate. Vo všetkých zvyšných modeloch budeme τ_r voliť ako 0.01.

Diferenciálne rovnice sme počítali numericky pomocou Crank-Nicolsonovej metódy. Crank-Nicolsonova metóda bola predstavená v [4] a je to implicitná metóda vznikajúca lineárnou kombináciou spätnej Eulerovej metódy a Eulerovej metódy napred so schémou:

$$y_{k+1} = y_k + rac{h}{2}(f(x_k, y_k) + f(x_{k+1}, y_{k+1})), \quad \mathrm{k} = 0, 1, 2, ...,$$
 (4.29)

kde x je premenná, podľa ktorej derivujeme (v našom prípade čas), y_k je vektor premenných (v našom prípade r a s), ktorý aproximuje skutočné riešenie systému diferenciálnych rovníc $y_k(x_k)$, h je dĺžka diskretizačného časového kroku a f je derivácia premenných. To, že je metóda implicitná znamená, že v predpise pre y_{k+1} sa nachádza $f(x_{k+1}, y_{k+1})$, čiže člen obsahujúci y_{k+1} . Takéto rovnice sa často nedajú riešiť analyticky. Na jej riešenie sme v každom kroku použili (v jazyku Python) funkciu fsolve z balíka scipy.optimize. Implicitnú metódu sme volili preto, lebo je stabilná a nezvykne divergovať (v neskorších simuláciách sa táto vlastnosť zíde). Navyše Crank-Nicolsonova metóda má oproti Eulerovej metóde napred (tiež implicitná metóda) menšiu diskretizačnú chybu.

Ešte jedna dôležitá vec je, že zmenou váh sa zjavne (podľa statických stavov) mení krivka r(s), čiže funkcia f. To je možno trochu neintuitívne, pretože funkcia fpredsa od W závislá nie je. My však máme na vodorovnej osi premennú, s a f berie ako argument Ws a navyše u nás platí r(s) = f(Ws). Nemení sa teda vstupnovýstupná funkcia f, ale jej argument. V niektorých zdrojoch zodpovedá vodorovnej osi priamo Ws. Vtedy sa so zmenou W mení práve oranžová krivka s(r) a nie modrá r(s). Model však ostáva úplne rovnaký, je to len zmena mierky.

4.5.2 Model bez synaptickej saturácie, synaptickej plasticity a s externým vstupom

Ak do jednotky bude prichádzať externý vstup (napr. ako dôsledok aktivity iných neurónov alebo pomocou elektródy), tak sa rovnica pre dr zmení na:

$$au_r rac{dr}{dt} = -r + f\left(\textit{W.s}(t) {+} s_{ ext{ext}}(t)
ight).$$

Takémuto modelu, bude prislúchať Obrázok 4.5.

Prvá simulácia na Obrázku 4.5 má rovnaké parametre ako prvá simulácia pri predošlom modeli (Obrázok 4.3). Simulácie na zvyšných riadkoch majú rovnaké parametre ako druhá simulácia z Obrázku 4.3.

Do jednotky bude prichádzať v čase od 400ms do 450ms externý vstup $s_{\text{ext}}(t)$. Jeho intenzita bude v rámci jednotlivých simulácií postupne 0.3, 0.3, 0.087, 0.075 a 0.000001.

V prvej simulácii na pravom grafe je vidno, že počas toho, čo prichádza do systému externý vstup, firing rate jednotky rastie. Po jeho "vypnutí" sa však firing rate jednotky (a samozrejme aj jej synaptický vstup) opäť vráti na svoju stabilnú hodnotu. Druhá až štvrtá simulácia veľmi dobre ukazujú už spomínanú stabilitu pevných bodov. Pokiaľ intenzitu externého vstupu zvolíme takú, že firing rate "vyskočí" do hodnôt vyšších ako sú stabilné pevné body systému (druhý riadok Obrázku 4.5), tak po jeho skončení firing rate po určitom čase klesne na hodnotu prislúchajúcu hodnote firing rate najvyššieho pevného bodu. Pri tretej a štvrtej simulácií externý vstup zvýši firing rate na hodnoty veľmi blízke nestabilnému pevnému bodu (sú to približne hodnoty, v ktorých sme začali názorné iterovanie vývoja s a r na Obrázku 4.4). Vidíme, že hodnota firing rate sa po skončení prichádzania externého vstupu nezačne približovať k najbližšiemu pevnému bodu



Obrázok 4.5 - Simulácie firing rate modelu s jednou jednotkou a s externým vstupom a bez zahrnutia synaptickej saturácie a synaptickej plasticity.

(ktorým je nestabilný stredný bod), ale začne sa od neho vzďaľovať k najbližšiemu stabilnému pevnému bodu, presne ako na Obrázku 4.4.

Takáto vlastnosť neurónov slúži ako pamäť, pretože neurón sa po príchode stimulu (externého vstupu) "prepne" do inej úrovne a ostane v nej aj po jeho skončení až do príchodu iného (napr. inhibičného) stimulu. Takéto neuróny sú preto v mozgu veľmi dôležité a nachádzajú sa v medzimozgovom hrbole (angl. *thalamus*) alebo mozgovej kôre.

Piata simulácia je niečo ako patologický prípad, pretože sme tentoraz nezvolili počiatočnú hodnotu firing rate ako r(t=0)=f(s=0), ale takú, aby prislúchala presne hodnote firing rate v nestabilnom pevnom bode (v tomto prípade je to 50). Vidíme, že pri neprítomnosti externého vstupu sa firing rate nemení, ale už aj extrémne malý synaptický vstup (v našej simulácii 0.000001) spôsobí vychýlenie, dôsledkom ktorého sa firing rate po určitom čase dostane na hodnotu prislúchajúcu nejakému stabilnému pevnému bodu⁴⁵.

Zaujímavé je aj to, že pri piatej simulácii sa z grafu môže zdať, že firing rate nezačne rásť hneď v čase začiatku príchodu externého vstupu, ktorým je 400ms, ale až v čase približne okolo 650ms, kedy už paradoxne externý vstup neprichádza. To však nie je pravda, firing rate začne rásť hneď v čase 400ms. Avšak vzhľadom na veľmi nízku hodnotu externého vstupu je tento nárast na grafe spočiatku nebadateľný. Firing rate však vzhľadom na tvar diferenciálnych rovníc, popisujúcich model, rastie exponenciálne, a preto sa po určitom čase táto minimálna zmena začne rapídne zväčšovať. Externý vstup slúži teda v piatej simulácii výhradne na vychýlenie a zvyšok už "spraví systém diferenciálnych rovníc sám".

⁴⁵ Všetky doterajšie úvahy o konvergencii firing rate do stabilného pevného bodu platia konkrétne pre náš model. Nemusí to takto pri diferenciálnych rovniciach platiť všeobecne. Systém môže napr. divergovať alebo oscilovať. Tak isto nemusí ani vždy platiť tvrdenie z predošlého odseku, že systém dokonverguje nutne k najbližšiem u pevnému bodu.

4.5.3 Model so synaptickou saturáciou a bez synaptickej plasticity

Pokiaľ do modelu zahrnieme aj synaptickú saturáciu, tak sa rovnica predszmení na:

$$rac{m{ au}_s}{1+m{ au}_s P_{
m rel} a_{m{0}} r(t)} rac{ds}{dt} = -s + rac{m{ au}_s P_{
m rel} a_{m{0}} r(t)}{1+m{ au}_s P_{
m rel} a_{m{0}} r(t)}.$$

 Pribudli nové konštanty: Pravdepodobnosť $P_{\rm rel},$ že vačok pri spiku uvoľní neurotransmiter a pravdepodobnosť a_0 , že sa otvorí chemicky riadený kanál, ak príde neurotransmiter. Je dôležité poznamenať, že to, že nám tieto pravdepodobnosti pribudli až teraz, neznamená, že v predošlých modeloch bez synaptickej saturácie sme ich neuvažovali. Ide skôr o to, že v predošlých modeloch sa tieto konštanty dali zakomponovať do konštanty W. Pri zahrnutí synaptickej saturácie sa tieto konštanty už vyskytujú v rovnici pre ds na viacerých miestach, a preto je vhodné ich do modelu zahŕňať explicitne. Budeme sa však pozerať iba na súčin týchto konštánt $P_{\rm rel}a_0$, ktorý interpretujeme ako pravdepodobnosť otvorenia chemicky riadeného kanálu v membráne postsynaptického neurónu, spôsobeného vygenerovaním spiku presynaptickým neurónom.

Vstupno-výstupná funkcia ostane rovnaká ako v predošlých modeloch a bránová funkcia bude mať tvar ako v (4.28).

Na Obrázku 4.6 môžeme vidieť výsledky simulácií. V prvom rade si môžeme všimnúť, že synaptická saturácia spôsobuje, že funkcia s(r) už nie je lineárna. V prvých dvoch simuláciách do systému neprichádza žiadny externý vstup, v tretej a štvrtej prichádza vstup výšky 0.8. Vo všetkých simuláciách sme zvolili W = 0.92 a $\tau_s = 0.05$. Konštanta τ_s určuje rýchlosť zatvárania sa chemicky riadených kanálov. Ak je príliš malá, tak sa kanály veľmi rýchlo zatvárajú. Saturácia, čiže stav, kedy sa nové kanály už neotvárajú (pretože väčšina je už otvorená), preto nemá pri nízkych hodnotách τ_s veľký vplyv. Aby sme ukázali vplyv saturácie, zvýšili sme τ_s na hodnotu 0.05 (oproti predošlým 0.002). Súčin pravdepodobností $P_{\rm rel}a_0$ sme v prvej a tretej simulácii zvolili ako 0.2 a v druhej a štvrtej ako 1. Pre $P_{\rm rel}a_0=0.2$ máme 3 pevné body a pre $P_{\rm rel}a_0=1$ máme jeden. Vidíme, že pri vyššej hodnote $P_{\rm rel}a_0$ je aj

hodnota firing rate v pevnom bode vyššia. To je logické, pretože čím vyššia $P_{\rm rel}a_0$, tým viac chemicky riadených kanálov každý spike otvorí a samoexcitačná jednotka je tak schopná aj bez externého vstupu udržiavať vyššie hodnoty firing rate.

V pravej časti vidíme to isté, čo pri predošlej simulácii, že vpustením externého prúdu do systému vieme "prepínat^{*}" stabilné úrovne firing rate. V prípade štvrtej simulácie je už pred príchodom externého vstupu firing rate takmer na svojom maxime, a preto ju externý kladný vstup už moc nezmení.



Obrázok 4.6 – Simulácie firing rate modelu s jednou jednotkou a s externým vstupom, so synaptickou saturáciou a bez synaptickej plasticity.

4.5.4 Model so synaptickou saturáciou, synaptickou depresiou a bez synaptickej facilitácie

Pri zahrnutí synaptickej plasticity bude už $P_{\rm rel}$ závislá od času a pribudnú aj diferenciálne rovnice pre D a F:

$$\begin{split} \tau_r \frac{dr}{dt} &= -r + f\left(W.s(t) + s_{\text{ext}}(t)\right), \\ \frac{\tau_s}{1 + \tau_s(p_0 DF) a_0 r(t)} \frac{ds}{dt} &= -s + \frac{\tau_s(p_0 DF) a_0 r(t)}{1 + \tau_s(p_0 DF) a_0 r(t)}, \\ \frac{dD}{dt} &= \frac{1 - D}{\tau_D} - (p_0 DF) r(t), \\ \frac{dF}{dt} &= \frac{1 - F}{\tau_F} + f_{\text{fac}} \cdot (1/p_0 - F) r(t). \end{split}$$

Najprv budeme uvažovať iba synaptickú depresiu. Na pripomenutie, synaptická depresia hovorí o tom, že na termináloch axónu je len obmedzený počet vačkov s neurotransmiterom. Čím intenzívnejšie je generovanie spikov, tým rýchlejšie tento počet klesá. Čím nižšia je konštanta τ_D , tým rýchlejšie sa vačky dopĺňajú. Synaptickú facilitáciu zatiaľ uvažovať nebudeme, čiže $f_{\text{fac}} = 0$, a teda F=1. τ_s zvolíme 0.002 a τ_r klasicky 0.01. Vstupno-výstupnú funkciu zvolíme ako $f(x) = \frac{100}{(1+e^{-10x+4.5})}$ a bránová funkcia je rovná stacionárnemu riešeniu pre ds. Výsledky simulácií sú zakreslené na Obrázku 4.7. V pravej časti obrázku máme tentoraz zakreslenú nie len krivku pre firing rate r, ale aj krivku pre premennú synaptickej depresie D. Nakoľko D je podiel pripravených vačkov na termináli axónu z ich maximálneho počtu, a teda číslo medzi 0 a 1, krivka pre D je pre vyššiu prehľadnosť vynásobená číslom 100. Každá simulácia zodpovedá určitej voľbe parametrov W, p_0, a_0 a τ_D a pri žiadnej neprichádza externý vstup. Parametre boli zvolené nasledovne:

	W	$p_0^{}$	a_{0}	$ au_D$
1. simulácia	77	0.65	0.65	0.6
2. simulácia	83	0.85	1	0.1
3. simulácia	71	0.5	0.5	0.5
4. simulácia	91	0.85	0.65	0.2
5. simulácia	97	0.85	0.85	0.6
6. simulácia	96	1	0.85	0.3

Tieto parametre neboli zvolené náhodne, ich význam však vysvetlíme neskôr.

Jeden z rozdielov oproti predošlým modelom je, že statické stavy (ľavá časť obrázku) už nemajú až takú veľkú výpovednú hodnotu, pretože sú teraz dané nie dvomi, ale tromi premennými. Vzhľadom na tvar diferenciálnych rovníc popisujúcich model je v 2D priestore možné zakresliť statický stav iba pre kombináciu s a r, nie pre D.

V pravej časti Obrázku 4.7 vidíme ďalšiu zásadnú zmenu oproti predošlým modelom. V predošlých modeloch bola firing rate pri neprítomnosti externého vstupu monotónna a vždy smerovala priamo k jej pevnému bodu. Tu je tomu tak iba v druhej a tretej simulácii. V ostatných simuláciách vidíme nemonotónnosť, niekedy dokonca oscilácie. Nemonotónnosť je spôsobená najmä časovými



Obrázok 4.7 – Simulácie firing rate modelu s jednou jednotkou, bez externého vstupu, so synaptickou saturáciou, synaptickou depresiou a bez synaptickej facilitácie.

konštantami, ktoré spomaľujú zmeny jednotlivých premenných a umožňujú tak nejakej premennej vyskočiť do vyšších hodnôt skôr ako iné "zareagujú". Ďalším zdrojom nemonotónnosti je fakt, že počiatočnú hodnotu premennej D volíme ako 1. T.j. všetky vačky sú pripravené. Otázkou je, či je pri jednotke s rekurentným excitačným spojením takáto voľba vhodná, matematicky je však prirodzená, keďže 1 je hodnota, ku ktorej premenná D pri neprítomnosti spikov konverguje.

Vidíme, že aj v tejto monotónnosti sú medzi jednotlivými simuláciami rozdiely. V prvej simulácii začne firing rate na začiatku rapídne stúpať, keď však podiel neurotransmiterov klesne na nízke hodnoty, firing rate znovu klesne a na nízkych hodnotách už obe premenné ostanú.

Vo štvrtej simulácii sa deje podobná vec, s tým rozdielom, že firing rate sa ustáli na relatívne vysokej hodnote. Jednotka s takýmito parametrami je teda schopná generovať relatívne vysoký počet spikov aj pri nízkom počte vačkov. Znova pripomíname, že sa jedná o jednotku iba s jedným, a to rekurentným spojením⁴⁶.

V piatej a šiestej simulácii nastávajú spomínané oscilácie. Rozdiel medzi nimi je však taký, že pri piatej simulácii sa po každom "zaoscilovaní" firing rate dlho spamätáva a "čaká" na doplnenie vačkov, následne opäť spraví jednu osciláciu a proces sa opakuje. V šiestej simulácii idú oscilácie plynule jedna za druhou s tým, že zmeny v D, ktoré tieto oscilácie spôsobujú, sú minimálne.

Keďže systém diferenciálnych rovníc popisujúci tento model je nelineárny a také systémy nezvyknú byť riešiteľné klasickými metódami (nie je ani jasné, či má systém vôbec riešenie v tvare elementárnych funkcií), nevieme presne popísať, pri akých hodnotách parametrov modelu nastáva daný priebeh firing rate.

Rozhodli sme sa preto použiť mierne netradičný prístup na preskúmanie vplyvu parametrov modelu na priebeh jeho riešenia.

⁴⁶ Tento fakt pripomíname v tejto podkapitole často, pretože vo zvyšných podkapitolách a kapitolách práce sme takéto spojenie nikdy neuvažovali a je veľmi dôležité, aby si čitateľ uvedomil, že v tomto prípade jednotka vkladá synaptický vstup sama sebe a sama ovplyvňuje svoju firing rate. To je veľký rozdiel oproti klasickým spojeniam, ktoré sme riešili v predošlých kapitolách.

V prvom kroku sme spustili veľké množstvo simulácií pre rôzne hodnoty parametrov W, p_0, a_0 a τ_D . Tieto hodnoty sme vyberali ako kombinácie hodnôt z nasledovných množín:

- $\circ \quad W \in \{40, \, 43, \, 46, \, 49, \, 52, \, 55, \, 58, \, 61, \, 64, \, 65, \, 66, \, 67, \, 68, \, 69, \, 70, \, 71, \, 72, \, 73, \, 74, \\ 75, \, 76, \, 77, \, 78, \, 79, \, 80, \, 81, \, 82, \, 83, \, 84, \, 85, \, 86, \, 87, \, 88, \, 89, \, 90, \, 91, \, 92, \, 93, \, 94, \, 95, \\ 96, \, 97, \, 98, \, 99, \, 100, \, 101, \, 102, \, 103, \, 104, \, 106, \, 109, \, 112\}$
- $\circ \quad p_0 \in \{0.5,\, 0.65,\, 0.85,\, 1\},$
- $\circ \quad a_0 \in \{0.5, \, 0.65, \, 0.85, \, 1\},$
- $\circ \quad \tau_D \in \{0.1, \, 0.15, \, 0.2, \, 0.3, \, 0.4, \, 0.5, \, 0.6, \, 0.7, \, 0.8, \, 1\}.$

Hodnoty boli vyberané tak, aby dávali zmysel po biologickej stránke a aby sme dosiahli čo najvyššiu variabilitu v priebehu firing rate, pretože napr. pri nízkych hodnotách kombinácie parametrov W, p_0, a_0 firing rate nikdy nenarastie do vyšších čísiel a dostávali by sme teda približne rovnaké výsledky bez ohľadu na to, koľko rôznych takýchto nízkych hodnôt by sme nasimulovali. Tento prípad napr. nastáva aj v našej 3. simulácii na Obrázku 4.7, kde sú hodnoty W, p_0, a_0 71, 0.5 a 0.5. Voľba nižších hodnôt teda výpovedný zmysel veľmi nemá. Pri veľmi vysokých hodnotách W by firing rate naopak vyskočila na 100 a ostala tam, tak ako v 2. simulácii. Preto ani veľmi vysoké hodnoty W nemajú ilustračný zmysel. Tak isto musí samozrejme platiť, že pravdepodobnosti $p_0 a a_0$ budú medzi 0 a 1, že časová konštanta τ_D bude kladná a že W bude kladné. Posledné podmienka nie je všeobecné pravidlo, ale u nás platiť musí, keďže v tomto príklade riešime excitačnú jednotku. Dostávame tak 52*10*4*4=8320 kombinácií parametrov, a teda aj 8320 simulácií a pre každú sme si uložili priebeh firing rate.

V druhom kroku sme z tohto uloženého priebehu vypočítali rôzne štatistiky. Štatistiky sme volili tak, aby čo najlepšie popisovali správanie firing rate. Počítali sme:

 Súčet absolútnych zmien firing rate. Táto štatistika rozdeľuje simulácie, v ktorých firing rate ostala nízka (tam budú zmeny veľmi minimálne), v ktorých firing rate narástla na vysokú hodnotu a ostala tam (tam budú zmeny okolo 100) a v ktorých oscilovala (tam budú zmeny vysoké).

- Standardná odchýlka. Hodnoty tejto štatistiky budú opäť nízke pre simulácie, v ktorých firing rate ostala nízka, ale tentoraz v simuláciách, v ktorých firing rate narástla na vysokú hodnotu a ostala tam, bude jej výška závisieť od toho, či firing narástla hneď alebo až neskôr. Tak isto pre simulácie, v ktorých firing rate začala oscilovať, bude veľkosť štandardnej odchýlky hovoriť o tom, aké veľké majú simulácie rozpätie.
- Minimum a maximum firing rate na intervale od 0.5s po 1s. Interval je skrátený z toho dôvodu, aby odzrkadľoval hodnoty firing rate po "ustálení", kedy už nemajú veľký vplyv počiatočné podmienky. Táto štatistika hovorí o tom, kde sa firing rate ustálila, či osciluje, a ak áno, tak ako veľmi.
- Najvyšší a najnižší pokles. Táto štatistika hovorí o "strmosti" rastu resp. poklesu firing rate.
- Počet lokálnych extrémov. Táto štatistika nie je úplne formálne správna, pretože nepočíta lokálne extrémy analyticky, ale len numericky, a to tak, že sa pozeráme na hodnoty, ktoré sú vyššie alebo nižšie aj od svojej predchádzajúcej, aj od nasledujúcej hodnoty.
- Počet oscilácií. Túto štatistiku sme počítali tak, že sme pre každú simuláciu vybrali nejakú hranicu h, pre ktorú sme spočítali počet "prekročení" zhora tejto hranice. Pod prekročením sa myslí postupnosť 2 hodnôt, z ktorých prvá je nad danou hranicou a druhá pod ňou. Hranicu h vyberáme ako h = r(2) 4sign(r(2) 50), kde r(2) je hodnota firing rate v čase 2s, čiže posledná hodnota firing rate na našom časovom úseku, a teda aj hodnota, do ktorej systém konverguje (ak konverguje) alebo hodnota, ktorá je v rozsahu oscilácií firing rate. Pripočítanie resp. odpočítanie 4 od hranice nám zaručuje, že nezapočítame oscilácie s veľmi nízkou amplitúdou okolo pevného bodu (aké sú napr. v 4. simulácii). Fakt, že vyberáme hodnotu r(2) zase zaručuje, že sa nestane, že firing rate bude oscilovať niekde nad alebo pod hranicou h.

V treťom kroku sme na naše simulácie aplikovali zhlukovací algoritmus. Do algoritmu vstupovali simulácie ako body v 8-rozmernom priestore, ktorých súradnice

boli hodnoty našich ôsmych štatistík. Na body sme tak, ako v podkapitole 2.3.4, aplikovali algoritmus k-means.

Je veľmi dôležité poznamenať, že ani zhlukovací algoritmus v tomto kroku, ani vypočítanie štatistík v predošlom, nie sú nutné kroky nášho postupu. Slúžia len na to, aby sme každej simulácii priradili označenie zodpovedajúce jej priebehu firing rate. Rovnako správne (dokonca presnejšie) by bolo pozrieť sa na každú simuláciu "ručne" a dať jej označenie podľa priebehu firing rate. Toto by však bolo vzhľadom na počet simulácií časovo veľmi náročné, a preto sme sa rozhodli vypočítať spomínané štatistiky a následne použiť zhlukovací algoritmus.

Práve z tohto dôvodu sme si dovolili výsledky zhlukovacieho algoritmu mierne upraviť, a to konkrétne tak, že sme zámerne preddefinovali vyšší počet zhlukov a následne sme niektoré zhluky spojili tak, aby sme dostali presne kategórie zodpovedajúce priebehom z Obrázku 4.7. Na Obrázku 4.7 sú vykreslené práve také simulácie, ktorých štatistiky boli najbližšie k centroidom daných zhlukov, teda niečo ako "typickí" zástupcovia jednotlivých kategórií.

Vo štvrtom kroku sme na simulácie aplikovali kategorizačný algoritmus náhodný strom (angl. *random tree*). Náhodný strom funguje tak, že postupne rozdeľuje priestor vysvetľovacích premenných rôznymi nadrovinami tak, aby na základe tohto rozdelenia vedel čo najlepšie určiť jednotlivé kategórie.

Ilustračný príklad funkcie náhodného stromu je ukázaný na nami vyrobenom Obrázku 4.8. Farby bodov na obrázku prislúchajú jednotlivým kategóriám. Strom v tomto príklade najprv rozdeľuje priestor pomocou priamky x=10. Tie premenné, ktoré majú hodnotu x menšiu ako 10 následne rozdelí podľa priamky y=8. Tie premenné, ktoré majú hodnotu x väčšiu ako 10 rozdelí podľa priamky y=15a následne tie z nich, ktoré majú hodnotu y väčšiu ako 6 rozdelí ešte pomocou x=16. V tomto ilustračnom prípade sú kategórie relatívne homogénne, a preto sa dajú dokonalo rozdeliť s použitím minimálneho počtu uzlov. V našom prípade to také jednoduché nie je.



Obrázok 4.8 – Ilustračný príklad náhodného stromu

Náš náhodný strom sme natrénovali na našich dátach – simuláciách, pričom vysvetľujúce premenné boli tentoraz nie vyššie spomínané štatistiky, ale parametre W, p_0, a_0 a τ_D a kategórie simulácií boli príslušnosti k zhlukom, čiže jednotlivé spôsoby priebehu firing rate.

Náhodný strom je klasifikačný algoritmus. My však klasifikovať nepotrebujeme, zaujímajú nás iba jednotlivé nadroviny, ktoré strom považuje za kľúčové pri separácií jednotlivých simulácií do kategórií. Chceme totižto zistiť, aké hodnoty parametrov vplývajú na priebeh firing rate v jednotlivých simuláciách.

Na Obrázku 4.9 vidíme výsledky algoritmu. Obdĺžniky zodpovedajú jednotlivým uzlom. V každom uzle je napísaná podmienka, ktorá sa v ňom posudzuje, koeficient neistoty *gini* [6], ktorý nám hovorí o tom, či je v danom uzle prevaha niektorej kategórie, alebo je tam stále veľká miera neistoty. Ďalej je tam celkový počet vzoriek (u nás sú vzorky jednotlivé simulácie) a počet vzoriek v jednotlivých kategóriách. Poradie kategórií zodpovedá poradiu kategórií na Obrázku 4.7, čiže prvá kategória (s celkovo 2498 vzorkami) sú simulácie, v ktorých mala firing rate taký priebeh, ako simulácia v prvom riadku Obrázku 4.7, druhá kategória (s celkovo 1291 vzorkami) sú simulácie, v ktorých mala firing rate taký priebeh, ako simulácie, v ktorých mala firing rate taký priebeh, ako simulácie, v ktorých mala firing rate taký priebeh, ako simulácia v prvom riadku Obrázku 4.7, druhá kategória (s celkovo 1291 vzorkami) sú simulácie, v ktorých mala firing rate taký

Takto sa dostávame k záveru nášho postupu a k dôvodu, prečo sme vlastne toto všetko robili. Z Obrázku 4.7 je totižto možné vyčítať viacero vecí:

- Prvá kategória má vo väčšine prípadov τ_D vyššie ako 0.25 (už v prvom uzle sa pri $\tau_D < 0.25$ zredukuje počet vzoriek z prvej kategórie z 2498 na 86). Tak isto sa vyznačuje a_0 nižšou ako 0.75, pretože zo 2498 vzoriek prvej kategórie v tomto uzle spĺňa $a_0 < 0.75$ 1769 vzoriek. Tak isto aj Wnižšou ako 68.5 pre tie simulácie, v ktorých $a_0 > 0.75$ (tam má W nižšie ako 68.5 až 422 vzoriek prvej kategórie z celkového počtu 643, pričom zo všetkých vzoriek v danom uzle to spĺňa len 728 z 2912). V prípade $a_0 < 0.75$ potom prevláda aj $\tau_D > 0.55$.
- Druhá kategória sa vyznačuje výrazne nízkym τ_D (v prvom uzle platí $\tau_D < 0.25$ pre 1289 vzoriek z druhej kategórie z celkového počtu 1291, a tak isto v prvom uzle na 3. úrovni platí $\tau_D < 0.125$ pre 283 vzoriek z druhej kategórie z 355). Tak isto sa vyznačuje a_0 vyššou ako 0.75 a Wvyššou ako 71.5.
- Tretia kategória nie je hodnotou τ_D veľmi ovplyvnená, je však veľmi silno ovplyvnená hodnotu a_0 . Podmienku $a_0 < 0.75$ spĺňa v "pravej" vetve 342 vzoriek z tretej kategórie z 364 a v "ľavej" 133 zo 140.
- Štvrtá kategória preferuje hodnoty τ_D medzi 0.25 a 0.125 (462 vzoriek z 569 spĺňa v prvom uzle τ_D<0.25, ale iba 70 z 270 spĺňa v prvom uzle na 3. úrovni τ_D<0.125). Iné premenné na túto kategóriu veľký vplyv podľa diagramu nemajú.
- Piata kategória sa rovnako ako prvá, vyznačuje τ_D vyšším ako 0.25. Na rozdiel od prvej však $\tau_D > 0.125$ veľmi nepreferuje (iba 172 z 801 vzoriek v danom uzle to spĺňa). Ďalší rozdiel medzi piatou a prvou kategóriu je, že prvá preferuje vyššie hodnoty a_0 a W.
- 22 vzoriek zo šiestej kategórie spadá do jednej z 3 kombinácií: (τ_D < 0.25 ∧ a₀ < 0.75 ∧ τ_D > 0.125) ∨ (τ_D < 0.25 ∧ a₀ > 0.75 ∧ W < 71.5) ∨ ∨ (τ_D > 0.25 ∧ a₀ > 0.75 ∧ W > 68.5)



Obrázok 4.9 – Schéme náhodného stromu pre našic
h8320 simulácií.

Klasifikačný strom na Obrázku 4.9 nie je úplný, pretože spodné "listy" (angl. *leaf*) nie sú homogénne, t.j. nie je v nich už len jedna kategória. Vrchné úrovne nám však dávajú najväčšiu informáciu (pretože algoritmus je postavený tak, aby vrchné úrovne najviac znižovali mieru neistoty *gini*) a pridávaním ďalších úrovní by sa stal strom neprehľadným.

Zaujímavosťou je aj to, že parameter p_0 sa nevyskytol v žiadnej nadrovine. Je veľmi pravdepodobné, že pri pridaní ďalších úrovní by sa vyskytol, ale dôvod, prečo nie je už v týchto prvých 3, je podľa nás to, že p_0 , ako pravdepodobnosť uvoľnenia pripraveného neurotransmitera, má síce pozitívny vplyv na s (pretože čím vyššie je p_0 , tým viac vačkov uvoľní neurotransmiter), ale má zároveň aj negatívny vplyv na D (podiel pripravených vačkov), pretože čím viac vačkov uvoľňuje neurotransmitery,

tým rýchlejšie ich podiel D klesá. Tieto dva vplyvy sa do istej miery navzájom kompenzujú, a preto sa domnievame, že parameter p_0 nevyšiel až tak signifikantný.

Záver je teda taký, že jeden "výstrel" firing rate na začiatku a následné zotrvanie v nízkych hodnotách tak, ako to máme v prvej simulácii z Obrázku 4.7, je spôsobené buď veľmi pomalým dopĺňaním vačkov s neurotransmiterom (čiže vysokým τ_D), alebo kombináciou relatívne pomalého dopĺňania vačkov s neurotransmiterom a slabou synaptickou váhou W. V prvom prípade by sa "výstrel" po nejakom čase zrejme zopakoval, v druhom je výstrel spôsobený iba vysokou počiatočnou hodnotou D a po jej "ustálení" firing rate dokonverguje k 0.

Pri druhej kategórii (druhej simulácii na Obrázku 4.7 zase vidíme, že na udržanie si konštante vysokej firing rate, je nutná výrazne vysoká rýchlosť dopĺňania vačkov (výrazne nízke τ_D), ideálne skombinovaná aj s vysokou synaptickou váhou W (to je hlavne preto, aby bol pevný bod firing rate vo vysokých hodnotách).

Pri priebehu firing rate ako v tretej simulácii firing rate nikdy nenarastie do vysokých hodnôt, je (ako bolo v tejto podkapitole už spomenuté) dôležité, aby mali synaptická váha W a pravdepodobnosť otvorenia kanálu a_0 (a aj p_0 , aj keď túto informáciu strom neposkytuje) dostatočne nízke hodnoty.

Priebeh firing rate, ako v piatej simulácii, pri ktorej nastávajú oscilácie s vysokou amplitúdou, ale s relatívne nízkou frekvenciou, nastáva pri spojení relatívne pomalého dopĺňania vačkov (to zapríčiňuje tú nízku frekvenciu) a vysokej synaptickej váhy W spolu s vysokou pravdepodobnosťou a_0 . Tie zapríčiňujú, že keď už sa D "dobije", tak je nárast firing rate veľmi výrazný.

Vysokofrekvenčné oscilácie, aké môžeme vidieť pri 4. a 6. simulácii, strom nevymedzil veľmi jednoznačne. Štvrtá simulácia, kde oscilácie relatívne rýchlo znižujú amplitúdu a konvergujú k hodnote niekde medzi 50 a 100, je zjavne veľmi podobná druhej, iba s tým rozdielom, že jeden z parametrov W,a_0 a τ_D nie je až taký vysoký.

Priebeh ako pri 6. simulácii, a teda vysokofrekvenčné oscilácie s približne konštantou amplitúdou, bez výraznejšieho náznaku konvergencie, je veľmi zriedkavý a zjavne vyžaduje skôr špeciálnu kombináciu parametrov ako nejaké jednoznačné pravidlo.

Na záver tejto podkapitoly ešte ukážeme, čo sa stane s našimi simuláciami z Obrázku 4.7 po pridaní externého vstupu v čase od 400ms do 750ms s hodnotou 0.1:



Obrázok 4.10 – Simulácie z Obrázku 4.7 po pridaní externého vstupu.

Z Obrázku 4.10 vidíme, že v prvej simulácií spôsobí pridanie externého vstupu ďalší "výstrel" firing rate. V druhej simulácii sa nezmení nič, nakoľko firing rate je už tak či tak na svojom maxime. V tretej simulácii vyvolá externý vstup tiež "výstrel" firing rate tak, ako v prvej. Vo štvrtej a šiestej simulácii externý vstup zastaví oscilovanie

a ustáli firing rate na nejakej, relatívne vysokej hodnote. V piatej simulácii sa zvýši frekvencia pomalých oscilácii a mierne sa zníži ich amplitúda.

Zatiaľ čo vplyv pridania externého vstupu s danou hodnotou je pri 2.,4. a 6. simulácii jasný, pri 1. a 3. sa naskytá otázka: "Budú sa "výstrely" opakovať?". Pri piatej je zase rozumné pýtať sa, či pri zvýšení intenzity externého vstupu nastane



rovnaký jav, ako pri 4. a 6., a teda zastavenie oscilácií a ustálenie na nejakej hodnote. Odpovede na tieto otázky môžeme vidieť na Obrázku 4.11.

Na Obrázku 4.11 je vykreslená už iba 1.,2. a 5. simulácia s tým, že pri 1. a 3. sme predĺžili externý vstup až po 2s a pri 5. sme zvýšili jeho intenzitu na 0.5. Vidíme, že

pri prvej simulácii pridanie externého vstupu o intenzite 0.1 naozaj začalo oscilovanie firing rate, podobné tomu z piatej simulácie na Obrázku 4.7. Pri tretej simulácii z Obrázku 4.10 (druhá na Obrázku 4.11) intenzita externého vstupu 0.1 nestačí na to, aby sa začalo oscilovanie a ten jeden "výstrel" je spôsobený iba tým, že v čase 400ms ešte podiel pripravených vačkov D nestihol dostatočne klesnúť. Nastáva teda priebeh ako pri prvej simulácii na Obrázku 4.7. Pri piatej simulácii (tretia na Obrázku 4.11) už zvýšená hodnota externého vstupu zabezpečí, že sa oscilovanie zastaví a firing rate sa ustáli na vysokej hodnote.

4.5.5 Model so synaptickou saturáciou a synaptickou facilitáciou

V nasledujúcej časti si ukážeme, čo sa stane s modelom, ak do neho zahrnieme namiesto synaptickej depresie synaptickú facilitáciu. Synaptická facilitácia hovorí o tom, že generovanie jednotlivých spikov môže na krátky čas po nich zvýšiť pravdepodobnosť vygenerovania ďalších. Miera, akou sa tá pravdepodobnosť zvyšuje, je daná konštantou $f_{\rm fac}$. Rýchlosť, akou sa pravdepodobnosť po zvýšení vráti do normálneho stavu, je daná konštantou τ_F .

Budeme používať rovnaký model, ako v predošlej časti. Tentoraz však bude konštanta $f_{\rm fac}$ nenulová. Synaptickú depresiu vynecháme tým, že položíme τ_D limitne blízke nule (to znamená, že vačky sa doplnia ihneď po vygenerovaní spiku). τ_s zvolíme 0.002 a τ_r klasicky 0.01. Vstupno-výstupná funkcia $f(x) = \frac{100}{(1+e^{-10x+2.5})}, a_0 =$ 0.5 a $p_0 = 0.1$. Voľba nízkeho p_0 je zámerná, pretože iba pri nízkom p_0 má synaptická facilitácia výrazný vplyv (pozri časť 4.3.2 a 4.3.3).

Premenné, ktoré budeme v jednotlivých simuláciách meniť, budú W, f_{fac} a τ_F . Ich hodnoty zvolíme postupne ako:

	W	$f_{ m fac}$	$ au_F$
1. simulácia	7.9	0.4	1.45
2. simulácia	7.9	0.15	0.65
3. simulácia	5.2	0.4	1.35
4. simulácia	7.7	0.1	0.5
5. simulácia	5.3	0.24	1.45

6. simulácia	6.7	0.1	1.15
7. simulácia	2	0.9	0.8

Do systému bude navyše vstupovať externý vstup o intenzite 0.1 v čase od 1
s po 1.1s. Výsledky simulácií môžeme vidieť na Obrázku 4.12. V pravej časti je tentoraz zaznačená okrem firing rate aj premenná F, prenásobená desiatimi (pripomíname, že premenná F nadobúda hodnoty z rozsahu od 1 po $1/p_0$).



Obrázok 4.12 – Simulácie firing rate modelu s jednou jednotkou, bez externého vstupu, so synaptickou saturáciou, synaptickou facilitáciou a bez synaptickej depresie.

Vidíme, že vzhľadom na pevné body systému prináša zahrnutie synaptickej facilitácie do modelu menšie zmeny, ako to bolo pri synaptickej depresii. Firing rate sa v neprítomnosti externého vstupu správa približne tak, ako tomu bolo v modeloch v častiach 4.5.1-4.5.3, t.j. konverguje k nejakému stabilnému pevnému bodu.

Rozdiel nastáva až po príchode a následnom odznení externého vstupu. V prvej simulácii máme len jeden pevný bod s vysokou hodnotou firing rate, preto systém sám dokonverguje do vysokých hodnôt a pridaný externý vstup efekt nemá.

Pri druhej simulácii máme 2 stabilné pevné body, podobne ako v časti 4.5.2. Firing rate je tak najprv ustálená v nižšom z nich. Príchod externého vstupu však zvýši jej hodnoty a firing rate následne dokonverguje k druhému, vyššiemu stabilnému pevnému bodu.

Tretia a piata simulácia sú veľmi podobné, rozdiel je však v tom, že pri tretej sa v ľavej časti modrá a oranžová krivka nedotýkajú, systém má teda len 1 pevný bod. Firing rate teda po odznení externého vstupu spadne do nízkych hodnôt. Vysoká hodnota τ_F však spôsobuje, že F klesá len veľmi pomaly, a preto sa firing rate veľmi dlho drží vo vyšších hodnotách. Pri piatej simulácii firing rate po odznení externého vstupu dokonverguje k hodnote okolo 75. Takáto hodnota pevného bodu je pri našom modeli veľmi netradičná a bez synaptickej facilitácie by nebola možná (spomeňme si, že v modeloch bez synaptickej facilitácie sa firing rate ustálila buď vo veľmi vysokých, alebo nízkych hodnotách). V tomto prípade je to možné práve preto, že premenná τ_F je vysoká a F je teda schopná držať si veľmi vysoké hodnoty aj pri nižších hodnotách firing rate.

Pri štvrtej simulácii, narastie firing rate vplyvom externého vstupu do vysokých hodnôt (až nad nestabilný pevný bod). Následne však nastáva netradičná situácia, kedy firing rate nezačne rásť k bližšiemu stabilnému pevnému bodu, ale poklesne späť k tomu nižšiemu. Tento priebeh nie je v súlade s priebehom popísanom na Obrázku 4.4. Tak isto nie je v súlade s tvrdením, že ak sa firing rate dostane do pevného bodu (hoci aj nestabilného), tak sa už ďalej nemení. V tomto prípade firing rate "prejde" nestabilným pevným bodom a pokračuje ďalej. Je spôsobené tým, že v statickom stave nie je zakreslená synaptická facilitácia. To, čo je statický stav pre premenné r a s, nemusí byť statický stav aj pre F. Preto, aj keď sú derivácie premenných r a s v pevnom bode nulové, zmena v F ich opäť vychýli (čo pre nestabilný pevný bod znamená, že sa systém začne od neho vzďaľovať). V šiestej simulácii začne firing rate po doznení externého vstupu klesať, ale facilitačná premenná Fnie. Nakoniec dosiahne Ftaké vysoké hodnoty, že firing začne znovu stúpať.

Pri poslednej simulácii nie je synaptická váha W dostatočne vysoká na to, aby externý vstup zvýšil firing rate do dostatočných hodnôt.

Hodnoty parametrov modelu v tejto časti sme vyberali podobným spôsobom, ako v predošlej časti (pomocou zhlukovania veľkého počtu simulácií). Cieľom bolo vybrať také kombinácie parametrov, aby boli simulácie čo najviac odlišné a aby čo najlepšie ukazovali vplyv synaptickej plasticity na model. Ako je však vidieť, synaptická facilitácia má menší vplyv na model ako synaptická depresia (simulácie so synaptickou facilitáciou sa viac podobajú na tie základné, bez akejkoľvek plasticity).

Na záver sme zahrnuli synaptickú facilitáciu do simulácií so synaptickou depresiou, ukázaných na Obrázku 4.7. Výsledky však neboli veľmi odlišné, preto ich neuvedieme. Dôvod je jednoduchý: Facilitačná premenná F je hodnota z intervalu <1, $1/p_0$ > a keďže hodnoty p_0 je v simuláciách z Obrázku 4.7 relatívne vysoká, synaptická facilitáciu by tu mala len veľmi minimálny efekt. Tento problém sme skúsili vyriešiť tak, že sme vydelili hodnoty p_0 štyrmi. Výsledky simulácií je možné vidieť na Obrázku 4.13. τ_F aj f_{fac} sme zvolili ako 0.8. Aj tu vidíme, že sa priebeh firing rate veľmi nezmenil oproti Obrázku 4.7. Potvrdzuje sa teda to, čo sme vyčítali z výsledkov náhodného stromu, že parameter p_0 pri synaptickej depresii nemá veľký vplyv na priebeh firing rate, a teda veľký vplyv nemá ani synaptická facilitácia, keďže tá ovplyvňuje model rovnakým spôsobom ako ho ovplyvňuje p_0 .



Obrázok 4.13 – Model zahŕňajúci aj synaptickú facilitáci
u aj synaptickú depresiu.

Záver

Cieľom práce bolo predstaviť niektoré matematické modely fungovania mozgu a neurónov a ilustrovať ich fungovanie prostredníctvom počítačových simulácií vybraných neurálnych aktivít.

V prvej kapitole sme vysvetlili základné pojmy ako mozog, neuróny a receptory. Vysvetlili sme akým spôsobom fungujú neuróny a ako si posielajú informáciu. Tak isto sme vysvetlili ako mozog tieto informácie spracováva.

V druhej kapitole sme sa venovali matematickým modelom popisujúcim neurálne kódovanie. Najprv sme predstavili základné matematické veličiny popisujúce reakciu jednotlivých neurónov, ako spike train alebo firing rate. Následne sme vysvetlili princíp stochastického generovania spike trainov a analyzovali sme rôzne spôsoby odhadnutia firing rate z nich. Predstavili sme kódovací model Linear-Nonlinear-Poisson model, spomenuli jeho možné komplikácie a navrhli vylepšenia.

V tretej kapitole sme sa zaoberali neurálnym dekódovaním. Popísali sme spôsoby, akými sa pri danej odpovedi neurónu modelujú pravdepodobnosti stimulu, ktorý ju vyvolal. Vysvetlili sme aj pojem ROC krivka a analyzovali jej vlastnosti na nami stochasticky nasimulovanými dátami. Tak isto sme ukázali využitie metódy maximálnej vierohodnosti pri odhadovaní najpravdepodobnejšieho stimulu pre populáciu neurónov.

V štvrtej kapitole sme sa zaoberali neurónovými sieťami, konkrétne firing rate modelom. Najprv sme vysvetlili princíp fungovania synapsií a spôsob, akým si neuróny cez ne odovzdávajú informácie. Následne sme predstavili pojmy ako synaptická vodivosť, synaptická saturácia a synaptická plasticita. Potom sme predstavili samotný firing rate model a diferenciálne rovnice, ktoré ho popisujú. Pokúsili sme sa objasniť všeobecný princíp tohto modelu a vysvetliť rozličné interpretácie jeho parametrov v rámci literatúry. Na záver sme analyzovali firing rate model s jednou skupinou neurónov (jednotkou) s rekurentnou odozvou. Postupne sme do modelu zahŕňali synaptickú saturáciu, synaptickú depresiu a synaptickú facilitáciu a pomocou numerických metód sme simulovali priebeh firing rate jednotky. Podrobne sme vysvetlili a ukázali vplyv parametrov na tento priebeh. Pri zložitejších variantoch modelu sme pri analýze parametrov využili metódy strojového učenia.

Hlavný prínos práce vidíme v prehľadnom spracovaní tejto rozsiahlej a zložitej tematiky. Pri popisovaní jednotlivých pojmov a modelov sme sa snažili kombinovať teoretické matematické odvodenia s vlastnými počítačovými simuláciami tak, aby o nich čitateľ nadobudol komplexný prehľad aj bez hlbších znalostí z oblasti Neurovedy. Navyše je pomerne málo slovenskej literatúry zameranej na túto tematiku a prínos tejto práce vidíme v jej rozšírení. Práve preto sme sa snažili v rámci tejto práce objasniť širšie spektrum podoblastí výpočtovej neurovedy.

Pokračovaním práce by mohlo byť rozšírenie firing rate modelu z poslednej podkapitoly aj na viac ako jednu jednotku a analyzovanie správania sa takejto siete jednotiek. Tak isto by bolo možné model rozšíriť aj o inhibičné spojenia medzi jednotkami a tiež popísať využitie informačnej teórie v modelovaní správania sa neurónov.
Zoznam použitej literatúry

 Britten, K. H., et al: The analysis of visual motion: a comparison of neuronal and psychophysical performance, Journal of Neuroscience 12.12 (1992), 4745-4765.

[2] Corrado, G. S., et al: *Linear-nonlinear-Poisson models of primate choice dynamics*, Journal of the experimental analysis of behavior 84.3 (2005), 581-617

[3] Coursera: *Computational Neuroscience*, dostupné na internete po zaregistrovaní sa do kurzu (16.5.2020):

https://www.coursera.org/learn/computational-neuroscience

 [4] Crank, J., Nicolson, P.: A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, Mathematical Proceedings of the Cambridge Philosophical Society Vol. 43. (1947), 50-67

[5] Daynan, P., Abott, L. F.: Theoretical Neuroscience, Massachusetts Institute of Technology Press, Boston, 2005

[6] Dorfman, R.: A formula for the Gini coefficient, The review of economics and statistics (1979), 146-149

[7] Frerking, M., Wilson, M.: Saturation of postsynaptic receptors at central synapses?, Current opinion in neurobiology 6.3 (1996), 395-403

[8] Harman, R., Rosa S.: Stochastické simulačné metódy, učebné texty, FMFI
 UK, Bratislava, 2019, dostupné na internete (16.5.2020):
 http://www.iam.fmph.uniba.sk/ospm/Harman/SSMp.pdf

[9] Jackman, S. L., Regehr, W. G.: *The mechanisms and functions of synaptic facilitation*, Neuron 94.3 (2017), 447-464

[10] King, J. R., et al: Encoding and decoding neuronal dynamics: Methodological framework to uncover the algorithms of cognition, preprint, HAL Archives-Ouvertes, 2018, dostupné na internete (17.05.2020): <u>https://hal.archives-ouvertes.fr/hal-01848442/</u>

[11] Kandel, E. R.: In Search of Memory: The Emergence of a New Science of Mind, W. W. Norton & Company, New York, 2007

[12] Martinez, L. M., et al.: *Receptive field structure varies with layer in the primary visual cortex*, Nature neuroscience 8.3 (2005), 372-379

[13] Miller, P.: An Introductory Course in Computational Neuroscience, Massachusetts Institute of Technology Press, Boston, 2018

[14] Nicholls, J. G.: Od neuronu k mozku, Academia, Praha, 2013

[15] Purves, D., Augustine, G. J., Fitzpatrick, D., Hall, W. C., LaMantia, A.
S., White, L. E.: *Neuroscience 5th edition*, Sinauer Associates, Sunderland, 2011

[16] Quiroga, R. Q., et al: Invariant visual representation by single neurons in the human brain, Nature 435.7045 (2005), 1102-1107

[17] Sejnowski, T. J., Koch, Ch., Churchland, P., S.: Computational neuroscience, Science 241.4871 (1988), 1299-1306 [18] Sherrington, Ch.: The integrative action of the nervous system, CUP Archive, Cambridge, 1952

 [19] Sun, Y., Beierlein, M.: Receptor saturation controls short-term synaptic plasticity at corticothalamic synapses, Journal of neurophysiology 105.5 (2011), 2319-2329

[20] The Human Memory: *Brain neurons and synapses*, dostupné na internete (17.5.2020):

https://human-memory.net/brain-neurons-synapses/

[21] Tibshirani R., Walther G., Hastie T.: *Estimating the number of clusters in a data set via the gap statistic*, Journal of the Royal Statistical Society Series B (Statistical Methodology) 63 (2001), 411-423, dostupné na internete (16.05.2020): https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00293

[22] Trappenberg, T.: Fundamentals of computational neuroscience, OUP, Oxford, 2009

[23] Trojan, S., Schreiber. M.: Knižní atlas biologie člověka: 430 modelových otázek k přijímacím zkouškám na medicínu+ 100 obrazových podkladů k opakování a procvičování, Scientia, Praha, 2007

[24] Van Steveninck, R. D. R., Bialek, W.: Real-time performance of a movementsensitive neuron in the blowfly visual system: coding and information transfer in short spike sequences, Proceedings of the Royal Society of London, Series B, Biological Sciences (1988), 379-414

[25] Weston, R.: F1.3YT2/YU3 Ordinary Differential Equations 1 & 2, učebné texty, Heriot-Watt University, 2008, dostupné na internete: http://www.macs.hw.ac.uk/~robertw/F13YT2/?fbclid=IwAR0rgTivP_zTU135ixT 6xAz0MsB0o9a7gnxZ_e1iS6A5QGJStIoNMHJUrow

[26] Widmaier, E. P., Raff, H., Strang, K. T.: Vander's Human Physiology 14th Edition, McGraw-Hill Education, New York, 2015

[27] Wikimedia Commons: *Neuron*, dostupné na internete (16.5.2020): https://commons.wikimedia.org/wiki/File:Neuron.svg

[28] Wikipedia: *Dirac delta function* dostupné na internete (16.5.2020): https://en.wikipedia.org/wiki/Dirac delta function

[29] Wikipedia: Human brain dostupné na internete (16.5.2020): https://en.wikipedia.org/wiki/Human_brain

[30] Wikipedia: *Point process* dostupné na internete (16.5.2020): <u>https://en.wikipedia.org/wiki/Point_process</u>

[31] Wikipedia: *Poisson point process* dostupné na internete (16.5.2020): <u>https://en.wikipedia.org/wiki/Poisson_point_process</u>

[32] Wikipedia: *White noise* dostupné na internete (16.5.2020): https://en.wikipedia.org/wiki/White_noise

[33] Wong, M., A., Hartigan, J., A.,: *A k-means clustering algorithm*, Journal of the Royal Statistical Society 28.1 (1979), 100-108.

[34] Zboja, S.: Neuróny a matematika, bakalárska práca, FMFI UK, Bratislava2018, dostupné na internete (14.1.2020):

 $\label{eq:http://opac.crzp.sk/?fn=detailBiblioForm\&sid=6AA8E1A8284D263B429E4E20F3\\98\&seo=CRZP-detail-kniha$

Prílohy

Zdrojové kódy v jazyku Python:

Simulácie z konca časti 2.2.2:

```
import numpy as np
from matplotlib import pyplot as plt
#### Vypocet viacerych spike trainov####
T = 2
M = 400000
delta_t = T/M
time = np.array(range(0,M))*delta_t
r = 4*(np.sin(4*time**2)+1)
n_{trials} = 200
spikes = np.zeros([n_trials,M])
for j in range(0,n_trials):
    for i in range(0,M):
        if (delta_t*r[i] >= np.random.uniform()):
                spikes[j,i] = 1
#### Vykreslenie firing rate####
plt.close()
plt.subplot(2, 1, 1)
plt.plot(time, r, label='r(t)')
plt.legend(loc="lower left")
#### Vypocet priemernej firing rate####
spikes avg = spikes.sum(axis=0)/n trials
r_mean = np.sum(spikes_avg)/T
#### Vykreslenie priemernej firing rate####
plt.subplot(2, 1, 1)
plt.plot(time, r_mean*np.ones(M), label='(r)')
plt.legend(loc="lower left")
#### Vykreslenie spike trainov####
pocet_pokusov_na_vykreslenie = 30
plt.subplot(2, 1, 2)
for j in range(0,pocet_pokusov_na_vykreslenie):
    for i in range(0,M):
        if spikes[j,i]==1:
            plt.vlines(time[i], j, j+0.9, colors='k', linestyles='solid')
plt.xlabel('cas')
plt.close()
r_hat = np.zeros(M)
#pomer velkosti w a T, cim viac tym mensie okno, max M min 1
hustota_delenia = 1
wt = (M/hustota_delenia)*delta_t
for i in range(0, int((wt/delta t)/2)):
    r_hat[i] = np.sum(spikes_avg[0:i+int((wt/delta_t)/2)])/((i+int((wt/delta_t)/2))*delt
a_t)
for i in range(int((wt/delta t)/2),M-int((wt/delta t)/2)+1):
    r hat[i] = np.sum(spikes avg[i-int((wt/delta t)/2):i+int((wt/delta t)/2)])/wt
for i in range(M-int((wt/delta_t)/2)+1,M):
    r_hat[i] = np.sum(spikes_avg[i-int((wt/delta_t)/2):M])/((M-
i+int((wt/delta t)/2))*delta t)
plt.subplot(4, 1, 1)
plt.plot(time, r, label='r(t)')
plt.legend(loc="lower left")
plt.subplot(4, 1, 1)
plt.plot(time, r_hat, label='r_hat(t) pri wt=2s')
```

```
plt.legend(loc="lower left", fontsize= 'xx-small')
plt.xticks([])
r hat = np.zeros(M)
#pomer velkosti w a T, cim viac tym mensie okno, max M min 1
hustota_delenia = 18.18
wt = (M/hustota delenia)*delta t
for i in range(0, int((wt/delta_t)/2)):
    r_hat[i] = np.sum(spikes_avg[0:i+int((wt/delta_t)/2)])/((i+int((wt/delta_t)/2))*delt
a_t)
for i in range(int((wt/delta t)/2),M-int((wt/delta t)/2)+1):
    r_hat[i] = np.sum(spikes_avg[i-int((wt/delta_t)/2):i+int((wt/delta_t)/2)])/wt
for i in range(M-int((wt/delta t)/2)+1,M):
    r_hat[i] = np.sum(spikes_avg[i-int((wt/delta_t)/2):M])/((M-
i+int((wt/delta t)/2))*delta t)
plt.subplot(4, 1, 2)
plt.plot(time, r, label='r(t)')
plt.legend(loc="lower left")
plt.subplot(4, 1, 2)
plt.plot(time, r_hat, label='r_hat(t) pri wt=110ms')
plt.legend(loc="lower left", fontsize= 'xx-small')
plt.xticks([])
r_hat = np.zeros(M)
#pomer velkosti w a T, cim viac tym mensie okno, max M min 1
hustota_delenia = 125
wt = (M/hustota_delenia)*delta t
for i in range(0, int((wt/delta_t)/2)):
    r_hat[i] = np.sum(spikes_avg[0:i+int((wt/delta_t)/2)])/((i+int((wt/delta_t)/2))*delt
a_t)
for i in range(int((wt/delta_t)/2),M-int((wt/delta_t)/2)+1):
    r_hat[i] = np.sum(spikes_avg[i-int((wt/delta_t)/2):i+int((wt/delta_t)/2)])/wt
for i in range(M-int((wt/delta t)/2)+1,M):
    r_hat[i] = np.sum(spikes_avg[i-int((wt/delta_t)/2):M])/((M-
i+int((wt/delta_t)/2))*delta_t)
plt.subplot(4, 1, 3)
plt.plot(time, r, label='r(t)')
plt.legend(loc="upper right", fontsize= 'x-small')
plt.subplot(4, 1, 3)
plt.plot(time, r_hat, label='r_hat(t) pri wt=16ms')
plt.legend(loc="upper right", fontsize= 'xx-small')
plt.xticks([])
r_hat = np.zeros(M)
#pomer velkosti w a T, cim viac tym mensie okno, max M min 1
hustota_delenia = 1000
wt = (M/hustota_delenia)*delta_t
for i in range(0, int((wt/delta_t)/2)):
    r_hat[i] = np.sum(spikes_avg[0:i+int((wt/delta_t)/2)])/((i+int((wt/delta_t)/2))*delt
a_t)
for i in range(int((wt/delta_t)/2),M-int((wt/delta_t)/2)+1):
    r_hat[i] = np.sum(spikes_avg[i-int((wt/delta_t)/2):i+int((wt/delta_t)/2)])/wt
for i in range(M-int((wt/delta_t)/2)+1,M):
    r hat[i] = np.sum(spikes avg[i-int((wt/delta t)/2):M])/((M-
i+int((wt/delta_t)/2))*delta_t)
plt.subplot(4, 1, 4)
plt.plot(time, r, label='r(t)')
plt.legend(loc="upper right", fontsize= 'x-small')
plt.subplot(4, 1, 4)
plt.plot(time, r_hat, label='r_hat(t) pri wt=2ms')
plt.legend(loc="upper right", fontsize= 'xx-small')
plt.xlabel('cas')
```

Počítanie STA v časti 2.3.2:

from __future__ import division
import numpy as np

```
import matplotlib.pyplot as plt
import pickle
# Nacitanie dat
data = 'c1p8.pickle'
with open(data, 'rb') as f:
    data = pickle.load(f)
stimul = data['stim']
odpoved = data['rho']
# Funkcia na pocitanie spike-triggered averageu
def compute sta(stimul, odpoved, tau max):
    sta = np.zeros((tau max,))
    spikes = odpoved[tau_max:].nonzero()[0] + tau_max
    pocet_spikov = len(spikes)
    for i in spikes:
        hodnoty_stimulu_pre_dany_spike = stimul[i-tau_max+1:i+1]
        sta = sta + hodnoty_stimulu_pre_dany_spike
    sta = sta/pocet_spikov
    return sta
# Casove parametre
jemnost_delenia_casovej_osi = 0.002
tau_max = 300
time = (np.arange(-tau_max, 0) + 1) * jemnost_delenia_casovej_osi
# Vypocet
sta = compute_sta(stimul, odpoved, tau_max)
# VYkreslenie
plt.plot(time, sta)
plt.xlabel('Time (s)')
plt.ylabel('Stimulus')
plt.title('Spike-Triggered Average')
plt.show()
```

```
Simulacie z 2.3.4
```

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
# pocet bodov
\#n = 1000000
n = 1000000
# ako daleko dozadu sa pozerame
tau max = 1000
dlzka 1 kroku = 10
pocet_deliacich_bodov = int(tau_max/dlzka_1_kroku)
time = np.arange(-tau_max, 0, dlzka_1_kroku)
# sablonv
stimul = np.random.randn(n)
od = -1
do = 1
x = np.arange(od,do,(do-od)/pocet_deliacich_bodov)
koef stupania = np.zeros(n-pocet deliacich bodov)
pom = np.zeros(n-pocet_deliacich_bodov)
spikes = np.zeros(n)
j=0
body = np.zeros([n, 101])
for i in range(pocet_deliacich_bodov,n):
    koef_stupania[i-pocet_deliacich_bodov] = np.sum((np.exp(x)-1)*stimul[i-
pocet_deliacich_bodov:i])
abskoef = np.abs(koef_stupania)
mk = np.max(abskoef)*2.5
randd = np.random.rand(n-pocet_deliacich_bodov)
# Generovanie spikov
```

```
for i in range(pocet deliacich bodov,n):
    pom[i-pocet deliacich bodov] = abskoef[i-pocet deliacich bodov]/mk
    #if ((koef_stupania[i-pocet_deliacich_bodov]>12) | (koef_stupania[i-
pocet_deliacich_bodov]<-12)):</pre>
   if (pom[i-pocet deliacich bodov]>randd[i-pocet deliacich bodov]):
        spikes[i] = 1
        body[j,0:100] = stimul[i-pocet_deliacich_bodov:i]
        if (koef_stupania[i-pocet_deliacich_bodov]>0):
            body[j, 100] = 1
        elif (koef stupania[i-pocet deliacich bodov]<0):</pre>
            body[j, 100] = 0
        j += 1
body = body[0:j,:]
print(len(body))
# pocitanie STA
STA = 0
for i in pd.Series(spikes)[pd.Series(spikes)==1].index:
    STA += stimul[i-pocet deliacich bodov:i]
STA = STA/len(pd.Series(spikes)[pd.Series(spikes)==1].index+100)
plt.plot(time, STA)
plt.xlabel('Cas (ms)')
plt.ylabel('Stimul')
plt.title('Spike-Triggered Average')
plt.ylim(-0.7,0.7)
plt.close()
# Vykreslenie toho, ako vyzeraju stimulu klesajuce a rastuce
for i in range(0,np.min([3000,len(body)])):
    if body[i,100]==1:
        plt.scatter(body[i,0],body[i,99],color='b')
    elif body[i,100]==0:
        plt.scatter(body[i,0],body[i,99],color='r')
plt.xlabel('1.zlozka', fontsize='medium')
plt.ylabel('100.zlozka', fontsize='medium')
plt.show()
plt.close()
# Laktovy diagram
SSD = []
K = range(1, 6)
for k in K :
    print(k)
    km = KMeans(n_clusters=k, init='k-
means++', max_iter=300, n_init=10, random_state=0)
    km = km.fit(body[:,0:99])
    SSD.append(km.inertia_)
plt.close()
plt.plot(K, SSD, 'bx-')
plt.xlabel('Počet clustrov')
plt.ylabel('Súčet štvorcov euklidovských vzdialeností bodov od centroidu')
plt.title('Lakťový diagram')
plt.show()
# clustrovaci algoritmus
kmeans = KMeans(n_clusters=2, init='k-means++', max_iter=300, n_init=10)
pred_y = kmeans.fit_predict(body[:,0:100])
print(np.sum(pred_y==body[:,100])/len(pred_y))
plt.close()
# Vykreslenie toho, ako vyzeraju stimulu klesajuce a rastuce
plt.subplot(2, 1, 1)
for i in range(0,np.min([3000,len(pred_y)])):
    if pred_y[i]==1:
        plt.scatter(body[i,0],body[i,99],color='c')
    elif pred_y[i]==0:
        plt.scatter(body[i,0],body[i,99],color='tab:orange')
plt.xlabel('1.zlozka', fontsize='medium')
plt.ylabel('100.zlozka', fontsize='medium')
plt.show()
# pocitanie vaicerych specialnych STA
```

```
STA1_1 = 0
STA1 \ 2 = 0
j = 0
prvy = pd.Series(pred_y)[pd.Series(pred_y)==1].index
druhy = pd.Series(pred_y)[pd.Series(pred_y)==0].index
for i in pd.Series(spikes)[pd.Series(spikes)==1].index:
    if j in prvy:
        STA1_1 += stimul[i-pocet_deliacich_bodov:i]
    elif j in druhy:
        STA1_2 += stimul[i-pocet_deliacich_bodov:i]
    j += 1
STA1_1 = STA1_1/len(prvy+100)
STA1_2 = STA1_2/len(druhy+100)
plt.subplot(2, 1, 2)
plt.plot(time, STA1_1, color='c')
plt.plot(time, STA1_2, color='tab:orange')
plt.xlabel('Cas (ms)')
plt.ylabel('Stimul')
plt.title('Spike-Triggered Average')
plt.ylim(-0.7,0.7)
plt.subplot(2, 1, 2)
```

ROC analyza z 3.3.3

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
# Generovanie vlastnych dat
r = np.arange(0, 3, 0.0001)
# suprema jednotlivych funkcii (c)
sup_minus = np.max((1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-1)**2)/(2*0.2**2)))
sup_plus1 = np.max((1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-
2)**2)/(2*0.2**2))/3+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-1)**2)/(2*0.2**2))/1.5)
sup_plus2 = np.max((1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-
2)**2)/(2*0.2**2))/3+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-1.2)**2)/(2*0.2**2))/1.5)
sup_plus3 = np.max((1/(0.3*np.sqrt(2*np.pi)))*np.exp(-((x-1.3)**2)/(2*0.3**2)))
sup_plus4 = np.max((1/(0.06*np.sqrt(2*np.pi)))*np.exp(-((x-
1)**2)/(2*0.06**2))*0.8+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-1.4)**2)/(2*0.2**2))/5)
# pocet vygenerovanych dat
n=15000
Xminus = []
while len(Xminus)<n:</pre>
   x = np.random.uniform()*5-1
    if sup_minus*(np.random.uniform())<((1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-</pre>
1)**2)/(2*0.2**2))):
        Xminus.append(x)
plt.hist(Xminus, np.linspace(0, 3, 100))
X1 = []
while len(X1)<n:</pre>
   x = np.random.uniform()*5-1
    if sup_plus1*(np.random.uniform())<((1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-</pre>
2)**2)/(2*0.2**2))/3+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-1)**2)/(2*0.2**2))/1.5):
       X1.append(x)
plt.hist(X1, np.linspace(0, 3, 100))
X2 = []
while len(X2)<n:</pre>
   x = np.random.uniform()*5-1
    if sup_plus2*(np.random.uniform())<((1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-</pre>
2)**2)/(2*0.2**2))/3+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-1.2)**2)/(2*0.2**2))/1.5):
        X2.append(x)
plt.hist(X2, np.linspace(0, 3, 100))
X3 = []
while len(X3)<n:</pre>
   x = np.random.uniform()*5-1
```

```
if sup_plus3*(np.random.uniform())<((1/(0.3*np.sqrt(2*np.pi)))*np.exp(-((x-</pre>
1.3)**2)/(2*0.3**2))):
        X3.append(x)
plt.hist(X3, np.linspace(0, 3, 100))
X4 = []
while len(X4)<n:</pre>
   x = np.random.uniform()*5-1
    if sup_plus1*(np.random.uniform())<((1/(0.06*np.sqrt(2*np.pi)))*np.exp(-((x-</pre>
1)**2)/(2*0.06**2))*0.8+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((x-
1.4)**2)/(2*0.2**2))/5):
        X4.append(x)
plt.hist(X4, np.linspace(0, 3, 100))
#Vypocet hranic z a ROC kriviek pre dane data
# jemnost delenia ciselnej osi na hladanie z
m=10000
ROC1_alpha = [0] * (m+1)
ROC1_beta = [0] * (m+1)
ROC2 beta = [0] * (m+1)
ROC3\_beta = [0] * (m+1)
ROC4_beta = [0] * (m+1)
spravne1 = 0
for i in range(0,m+1):
    ROC1_alpha[i] = np.sum(pd.Series(Xminus)>i*3/m)/len(Xminus)
    ROC1_beta[i] = np.sum(pd.Series(X1)>i*3/m)/len(X1)
    if spravne1 < np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X1)>i*3/m):
        spravne1 = np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X1)>i*3/m)
        spravne1_idx = i*3/m
spravne2 = 0
for i in range(0,m+1):
    ROC2_beta[i] = np.sum(pd.Series(X2)>i*3/m)/len(X2)
    if spravne2 < np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X2)>i*3/m):
        spravne2 = np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X2)>i*3/m)
        spravne2 idx = i*3/m
spravne3 = 0
for i in range(0,m+1):
    ROC3 beta[i] = np.sum(pd.Series(X3)>i*3/m)/len(X3)
    if spravne3 < np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X3)>i*3/m):
        spravne3 = np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X3)>i*3/m)
        spravne3 idx = i*3/m
spravne4 = 0
for i in range(0,m+1):
    ROC4_beta[i] = np.sum(pd.Series(X4)>i*3/m)/len(X4)
    if spravne4 < np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X4)>i*3/m):
        spravne4 = np.sum(pd.Series(Xminus)<i*3/m)+np.sum(pd.Series(X4)>i*3/m)
        spravne4_idx = i*3/m
ROC4_alpha = ROC1_alpha
ROC3_alpha = ROC1_alpha
ROC2_alpha = ROC1_alpha
plt.close()
velkost = 'medium'
# Vykreslenie hustot z ktorych sme generovali data
plt.subplot(4, 3, 1)
plt.plot(r, (1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-1)**2)/(2*0.2**2)), label=r'$f_{-
}\left(x\right)$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.subplot(4, 3, 1)
plt.plot(r, (1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-
2)**2)/(2*0.2**2))/3+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-
1)**2)/(2*0.2**2))/1.5, label=r'$f_{+_{1}}\left(x\right)$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.title(r'Husoty z ktorych sme generovali data', fontdict=None, fontsize= velkost, loc
='center', pad=None)
plt.xlabel('r', fontsize='medium')
plt.subplot(4, 3, 4)
plt.plot(r, (1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-1)**2)/(2*0.2**2)), label=r'$f_{-
}\left(x\right)$')
```

```
plt.legend(loc="upper right", fontsize= 'x-large')
plt.subplot(4, 3, 4)
plt.plot(r, (1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-
2)**2)/(2*0.2**2))/3+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-
1.2)**2)/(2*0.2**2))/1.5, label=r'$f_{+_{2}}\left(x\right)$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.xlabel('r', fontsize='medium')
plt.subplot(4, 3, 7)
plt.plot(r, (1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-1)**2)/(2*0.2**2)), label=r'$f_{-
}\left(x\right)$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.subplot(4, 3, 7)
plt.plot(r, (1/(0.3*np.sqrt(2*np.pi)))*np.exp(-((r-
1.3)**2)/(2*0.3**2)), label=r'$f_{+_{3}}\left(x\right)$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.xlabel('r', fontsize='medium')
plt.subplot(4, 3, 10)
plt.plot(r, (1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-1)**2)/(2*0.2**2)), label=r'$f_{-
}\left(x\right)$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.subplot(4, 3, 10)
plt.plot(r, (1/(0.06*np.sqrt(2*np.pi)))*np.exp(-((r-
1)**2)/(2*0.06**2))*0.8+(1/(0.2*np.sqrt(2*np.pi)))*np.exp(-((r-
1.4)**2)/(2*0.2**2))/5, label=r'$f_{+_{4}}\left(x\right)$'
plt.legend(loc="upper right", fontsize= 'x-large')
plt.xlabel('r', fontsize='medium')
# Vykreslenie histogramov vygenerovanych dat
plt.subplot(4, 3, 2)
plt.hist(Xminus, np.linspace(0, 3, 100))
plt.subplot(4, 3, 2)
plt.hist(X1, np.linspace(0, 3, 100))
plt.subplot(4, 3, 2)
plt.vlines(spravne1_idx, 0, 900, label='$z$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.title(r'Vykreslenie vygenerovanych dat a optimalna hranica $z$', fontdict=None, font
size= velkost, loc='center', pad=None)
plt.xlabel('r', fontsize='medium')
plt.ylabel('pocet', fontsize='medium')
plt.subplot(4, 3, 5)
plt.hist(Xminus, np.linspace(0, 3, 100))
plt.subplot(4, 3, 5)
plt.hist(X2, np.linspace(0, 3, 100))
plt.subplot(4, 3, 5)
plt.vlines(spravne2_idx, 0, 900, label='$z$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.xlabel('r', fontsize='medium')
plt.ylabel('pocet', fontsize='medium')
plt.subplot(4, 3, 8)
plt.hist(Xminus, np.linspace(0, 3, 100))
plt.subplot(4, 3, 8)
plt.hist(X3, np.linspace(0, 3, 100))
plt.subplot(4, 3, 8)
plt.vlines(spravne3_idx, 0, 900, label='$z$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.xlabel('r', fontsize='medium')
plt.ylabel('pocet', fontsize='medium')
plt.subplot(4, 3, 11)
plt.hist(Xminus, np.linspace(0, 3, 100))
plt.subplot(4, 3, 11)
plt.hist(X4, np.linspace(0, 3, 100))
plt.subplot(4, 3, 11)
plt.vlines(spravne4_idx, 0, 1300, label='$z$')
plt.legend(loc="upper right", fontsize= 'x-large')
plt.xlabel('r', fontsize='medium')
plt.ylabel('pocet', fontsize='medium')
# Vykreslenie ROC kriviek
```

```
plt.subplot(4, 3, 3)
plt.plot(ROC1 alpha, ROC1 beta)
plt.title('ROC krivka', fontdict=None, fontsize= velkost, loc='center', pad=None)
plt.scatter(ROC1_alpha[int(spravne1_idx*m/3)],ROC1_beta[int(spravne1_idx*m/3)])
plt.xlabel('alpha', fontsize='medium')
plt.ylabel('beta', fontsize='medium'
plt.subplot(4, 3, 6)
plt.plot(ROC2_alpha, ROC2_beta)
plt.scatter(ROC2_alpha[int(spravne2_idx*m/3)],ROC2_beta[int(spravne2_idx*m/3)])
plt.xlabel('alpha', fontsize='medium')
plt.ylabel('beta', fontsize='medium')
plt.subplot(4, 3, 9)
plt.plot(ROC3_alpha, ROC3_beta)
plt.scatter(ROC3_alpha[int(spravne3_idx*m/3)],ROC3_beta[int(spravne3_idx*m/3)])
plt.xlabel('alpha', fontsize='medium')
plt.ylabel('beta', fontsize='medium')
plt.subplot(4, 3, 12)
plt.plot(ROC4 alpha, ROC4 beta)
plt.scatter(ROC4_alpha[int(spravne4_idx*m/3)],ROC4_beta[int(spravne4_idx*m/3)])
plt.xlabel('alpha', fontsize='medium')
plt.ylabel('beta', fontsize='medium')
print(spravne1/30000*100)
print(spravne2/30000*100)
print(spravne3/30000*100)
print(spravne4/30000*100)
```

Simulácia z Obrázku 4.12 – ostatné simulácie v podkapitole 4.5 sú na rovnaký spôsob

```
import numpy as np
from matplotlib import pyplot as plt
from scipy.optimize import fsolve
%matplotlib qt
# Parametre 3
W_list =
               [77, 83, 71, 91, 97, 96]
# pravdepodobnosti 0,1
p0 list =
              [0.65, 0.85, 0.5, 0.85, 0.85, 1]
alpha_0_list = [0.65, 1, 0.5, 0.65, 0.85, 0.85]
# facilit
fac_list = [0.8]*len(W_list)
tau_f_list = [0.8]*len(W_list)
# depres 2
tau d list = [0.6, 0.1, 0.5, 0.2, 0.6, 0.3]
# externy
s_ext_sila = [0]*len(W_list)
tau s list = [0.002]*len(W list)
tau r_list = [0.01]*len(W_list)
# Vstupno-vystupna funkcia a zaroven staticka pre r
def f(x):
    return 100/(1 + np.exp(-(x-0.45)/0.1))
r_max = np.max(f(np.arange(0,100,0.01)))
# Branova funkcia a zaroven staticka pre s
def Fgate(r,D,F):
    return r*(p0*D*F)*alpha_0*(tau_s/(1+r*tau_s*(p0*D*F)*alpha_0))
def drsDF(r,s,D,F,s_ext):
    return np.array([-r/tau_r + f(W*s+s_ext)/tau_r,-
s/(tau_s/(1+r*tau_s*(p0*D*F)*alpha_0)) + Fgate(r,D,F)/(tau_s/(1+r*tau_s*(p0*D*F)*alpha_0
)),(1-D)/tau_d - (p0*D*F)*r,(1-F)/tau_f + fac*(1/p0-F)*r])
# staticka rovnica pre F
def Fstat(r):
    return (p0+fac*r*tau_f)/(p0+p0*fac*r*tau_f)
# staticka rovnica pre D
def Dstat(r,F):
    return 1/(1+p0*r*F*tau_d)
def F2(p):
```

```
x, y, z, zz = p
    return \
    (x - vsetky[0,j-1] - t_step/2 * (drsDF(vsetky[0,j-1],vsetky[1,j-1],vsetky[2,j-
1],vsetky[3,j-1],s_ext[j-1])[0] + drsDF(x,y,z,zz,s_ext[j])[0]),\
    y - vsetky[1,j-1] - t_step/2 * (drsDF(vsetky[0,j-1],vsetky[1,j-1],vsetky[2,j-
1],vsetky[3,j-1],s_ext[j-1])[1] + drsDF(x,y,z,zz,s_ext[j])[1]),\
    z - vsetky[2,j-1] - t_step/2 * (drsDF(vsetky[0,j-1],vsetky[1,j-1],vsetky[2,j-
1],vsetky[3,j-1],s_ext[j-1])[2] + drsDF(x,y,z,zz,s_ext[j])[2]),\
    zz - vsetky[3,j-1] - t_step/2 * (drsDF(vsetky[0,j-1],vsetky[1,j-1],vsetky[2,j-
1],vsetky[3,j-1],s_ext[j-1])[3] + drsDF(x,y,z,zz,s_ext[j])[3]))
# Casovy krok
t step = 0.001
# Dlzka pokusu
t max = 2
# Casova os
t = np.arange(0,t_max,t_step)
vsetky = np.array([np.zeros(len(t)),np.zeros(len(t)),np.zeros(len(t)),np.zeros(len(t))])
# Vykreslenie
# Synapticky vstup os
s_os = np.arange(0,0.018,0.0001)
# Firing rate os
r_{os} = np.arange(0, r_{max}, 0.0001)
fig, ax = plt.subplots(figsize=(12, 12))
for i in range(0,len(W_list)):
    W = W list[i]
    alpha_0 = alpha_0_list[i]
    tau_s = tau_s_list[i]
   tau_r = tau_r_list[i]
tau_f = tau_f_list[i]
   tau d = tau d list[i]
    fac = fac list[i]
    p0 = p0_{list[i]/4}
    # Externy prud
    s ext = np.zeros(len(t))
    plt.subplot(len(W_list), 2, 2*i+1)
   # Staticky stav
    plt.plot(s_os, f(W*s_os), label=r'$r(s)$')
    plt.plot(Fgate(r_os,Dstat(r_os,Fstat(r_os)),Fstat(r_os)), r_os, label=r'$s(r)$')
   plt.legend(loc='lower right', fontsize= 'small')
plt.xlabel('synapticky vstup')
    plt.ylabel('firing rate')
   # Vyvoj firing rate v case
    plt.subplot(len(W_list), 2, 2*i+2)
   # pociatocne podmienky
    vsetky[1,0] = 0
   vsetky[0,0] = f(W*vsetky[1,0])
    vsetky[2,0] = Fstat(vsetky[1,0])
    vsetky[3,0] = Dstat(vsetky[1,0],vsetky[2,0] )
    # Numericky vypocet diferencialnych rovnic pomocou metody Crank Nicolson
  for j in range(1,len(t)):
        vsetky[:,j] = fsolve(F2, (vsetky[0,j-1],vsetky[1,j-1],vsetky[2,j-1],vsetky[3,j-
1]))
    plt.plot(t,vsetky[0,:], label=r'$r(t)$')
    plt.plot(t,vsetky[2,:]*100, label=r'$100*D(t)$', color='r')
    plt.plot(t,vsetky[3,:]*10, label=r'$10*F(t)$', color='g')
    plt.ylim((0, r_max+10))
    plt.legend(loc='lower right', fontsize= 'small')
    plt.xlabel('cas')
    plt.ylabel('firing rate')
plt.subplot(len(W_list), 2, 2)
plt.title(r'Priebeh firing rate,synaptickej depresie a synaptickej facilitácie v case be
z externeho vstupu', fontdict=None, fontsize= 'small', loc='center', pad=None)
plt.subplot(len(W_list), 2, 1)
plt.title(r'Hodnoty statickeho stavu pre $r(s)$ a $s(r)$', fontdict=None, fontsize= 'med
ium', loc='center', pad=None)
```