

Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота магістра

на тему: «Дослідження моделей мереж нейронів»

Виконав: студент групи К23-1М
Спеціальність 122 «Комп'ютерні науки»

Комишний Кирило Васильович
(прізвище та ініціали)

Керівник к.ф.-м.н., доц. Фірсов О.Д.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент ДДТУ
(місто роботи)

в.о. завідувача кафедри програмного
забезпечення систем
(посада)

к.т.н., доцент Жульковський О. О.
(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2025

АНОТАЦІЯ

Комишний К.В. Дослідження моделей мереж нейронів

Дипломна робота на здобуття освітнього ступеня магістр за спеціальністю 122 «Комп'ютерні науки» – Університет митної справи та фінансів, Дніпро, 2025.

Зв'язки між біологією і математикою стали вже цілком звичними. Це відноситься перш за все до генетики і дослідження динаміки популяцій. Поки що важко говорити про математичну біологію в цілому як про науку, що склалася, але математична генетика вже поза сумнівом стала дисципліною, що достатньо сформувалася. Так само і в дослідження взаємостосунків між популяціями тварин, створюючи співтовариство, у вивчення динаміки чисельності популяцій математичні методи міцно ввійшли ще декілька десятиліть тому. Нарешті для всієї біології в цілому стало вже традиційним широке застосування математичної статистики, різних методів математичної обробки результатів експерименту.

Окреме місце в математичних дослідженнях займає проблема створення штучного інтелекту. Вчені у всьому світі дійшли висновку, що для створення ефективної моделі штучного інтелекту, працюючої за тими ж принципами, що і людський мозок, необхідно детально розібратися в принципах передачі, зберігання і обробки інформації в корі головного мозку ссавців.

Однією з найважливіших проблем при моделюванні систем штучного інтелекту є побудова моделей функціонування кори головного мозку ссавців.

Актуальним є дослідження механізмів генерації імпульсів збудження нейронів кори головного мозку ссавців, які мають форму спайків та залпів; дослідження математичних моделей нейронів спеціальних видів.

Потрібно програмно реалізувати моделі спайкових та залпових нейронів, провести імітаційне моделювання активності цих нейронів.

Також необхідно виділити значущі біологічні та нейрокомп'ютерні властивості моделей нейронів спеціальних видів, розрахувати обчислювальну складність програмної реалізації моделей нейронів.

Ключові слова: нейрон, модель, обчислення, обчислювальні операції.

ABSTRACT

Komyshny K.V. Research on the model net of neurons

Thesis for the degree of Master in specialty 122 "Computer Science" - University of Customs and Finance, Dnipro, 2025.

The connections between biology and mathematics have already become quite familiar. This applies primarily to genetics and the study of population dynamics. It is still difficult to speak of mathematical biology as a whole as an established science, but mathematical genetics has undoubtedly become a sufficiently formed discipline. Similarly, mathematical methods have firmly entered the study of the relationships between animal populations that form a community, and the study of population dynamics several decades ago. Finally, for all of biology as a whole, the widespread use of mathematical statistics and various methods of mathematical processing of experimental results has become traditional.

A separate place in mathematical research is occupied by the problem of creating artificial intelligence. Scientists around the world have concluded that in order to create an effective model of artificial intelligence that works on the same principles as the human brain, it is necessary to understand in detail the principles of information transmission, storage and processing in the cerebral cortex of mammals. One of the most important problems in modeling artificial intelligence systems is building models of the functioning of the cerebral cortex of mammals.

The study of the mechanisms of generation of excitation pulses of neurons of the cerebral cortex of mammals, which have the form of spikes and volleys, is relevant; the study of mathematical models of neurons of special species.

It is necessary to programmatically implement models of spike and volley neurons, conduct simulation modeling of the activity of these neurons.

It is also necessary to highlight significant biological and neurocomputing properties of models of neurons of special species, calculate the computational complexity of the software implementation of neuron models.

Keywords: neuron, model, calculations, computational operations.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Роль математичних моделей в біологічних дослідженнях.....	9
1.2. Аксиоматичні моделі активних середовищ.....	10
1.2.1. Аксиоматика моделі.....	10
1.2.2. Динамічна пам'ять.....	11
1.2.3. Відносна рефрактерність.....	12
1.3. Іонно-мембранні механізми збудження.....	14
1.4. Моделі поширення збудження.....	17
1.5. Обчислювальні можливості нейронів.....	21
1.6. Висновки до першого розділу.....	22
РОЗДІЛ 2. КЛАСИЧНІ ОБЧИСЛЮВАЛЬНІ АРХІТЕКТУРИ.....	23
2.1. Архітектура фон Неймана.....	23
2.2. DSP-процесори.....	24
2.3. Нейронні мережі.....	29
2.4. Висновки до другого розділу.....	34
РОЗДІЛ 3. МОДЕЛІ НЕЙРОНІВ.....	35
3.1. Еквівалентна схема мембрани гігантського аксона кальмара.....	35
3.2. Спайкові моделі.....	37
3.3. Модель Ходжкіна-Хакслі.....	41
3.4. Нейрокомп'ютерні властивості нейронів.....	41
3.4. Висновки до третього розділу.....	47
4. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ МЕРЕЖІ НЕЙРОНІВ.....	48
4.1. Порівняння моделей нейронів спеціальних видів за параметрами.....	48
4.2. Імітування активності нейронів.....	53
4.3. Висновки до четвертого розділу.....	62
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
ДОДАТОК А.....	68

ВСТУП

Для сучасної науки характерне застосування точних математичних методів в самих різних областях. Ці методи проникають в економіку, лінгвістику, психологію і багато інших областей знання, у тому числі і в біологію. У науку про живу природу математика входить різними шляхами: з одного боку — це використання сучасної обчислювальної техніки для швидкої обробки результатів біологічного експерименту, з іншої — створення математичних моделей, що описують різні живі системи і процеси, що відбуваються в них. Не менш важливий і «зворотний зв'язок», що виникає між математикою і біологією: біологія не тільки служить ареною для застосування математичних методів, але і стає все більш істотним джерелом нових математичних задач.

Деякі зв'язки між біологією і математикою стали вже цілком звичними. Це відноситься перш за все до генетики і дослідження динаміки популяцій. Якщо поки що важко говорити про математичну біологію в цілому як про науку, що склалася, то математична генетика вже поза сумнівом стала дисципліною, що достатньо сформувалася. Так само і в дослідження взаємостосунків між популяціями тварин, створюючими співтовариство, у вивчення динаміки чисельності популяцій математичні методи міцно ввійшли ще декілька десятиліть тому. Нарешті для всієї біології в цілому стало вже традиційним широке застосування математичної статистики, різних методів математичної обробки результатів експерименту.

Окреме місце в математичних дослідженнях займає проблема створення штучного інтелекту. Зараз можна бачити сполох інтересу до штучного інтелекту, який побудовано на системах лексичних аналізаторів. Але така технологія на є справжнім штучним інтелектом і не дозволяє отримати принципово нові знання чи інструменти для вирішення класичних чи нестандартних задач. Отже вчені у всьому світі дійшли висновку, що для створення ефективної моделі штучного інтелекту, працюючої за тими ж

принципами, що і людський мозок, необхідно детально розібратися в принципах передачі, зберігання і обробки інформації в корі головного мозку ссавців.

Однією з найважливіших проблем при моделюванні систем штучного інтелекту є побудова моделей функціонування кори головного мозку ссавців, як максимально доступної для аналізу системи.

Актуальним є дослідження механізмів генерації імпульсів збудження нейронів кори головного мозку ссавців, які мають форму спайків та залпів; дослідження математичних моделей нейронів спеціальних видів, функціонування їх мереж.

Потрібно програмно чи апаратно реалізувати моделі спайкових та залпових нейронів, провести імітаційне моделювання активності цих нейронів.

Також необхідно виділити значущі біологічні та нейрокомп'ютерні властивості моделей нейронів спеціальних видів, розрахувати обчислювальну складність програмної реалізації моделей нейронів і відповідно мереж, що побудованні з цих нейронів.

Побудова імітаційної нейронної мережі, складеної зі спайкових нейронів наблизить вчених до створення системи штучного інтелекту.

Магістерська робота присвячена застосуванню моделей нейронів для побудови моделі мережі нейронів та дослідження її параметрів.

Метою роботи є аналіз і обґрунтування технології організації обчислювального процесу для виконання спеціального алгоритма розв'язання обчислювальних задач.

Об'єктом дослідження у роботі є моделі нейронів та модель мережі нейронів.

Предметом дослідження є параметри мережі нейронів та обчислювальний процес її реалізації.

Методи дослідження, що використовуються у магістерській роботі – це безсерверні хмарні обчислення, паралельні обчислення, математичне моделювання.

Задачі дослідження – розробити, протестувати технологію моделювання, яка дозволить за допомогою обчислювального експерименту підтвердити емпіричні параметри існуючої моделі, провести обчислювальний експеримент, проаналізувати результати.

Структура магістерської роботи: робота складається з ___ розділів, об'єм роботи – ___ сторінок, робота містить ___ рисунків, перелік використаних джерел має ___ посилань.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Роль математичних моделей в біологічних дослідженнях

Кажучи про роль моделей в біологічних дослідженнях, важливо мати на увазі наступне. Хоча термін «модель» ми розуміємо в абстрактному значенні — як деяку систему логічних понять, а не як реальний фізичний пристрій, все ж таки модель — це щось істотно більше, ніж простий опис явища або чисто якісна гіпотеза, в якій ще залишається досить місця для різного роду незрозумілостей і суб'єктивних думок. Нагадаю приклад, що відноситься до досить далекого минулого. Свого часу Гельмгольц, займаючись вивченням слуху, висунув так звану резонансну теорію, що виглядала правдоподібно з чисто якісної сторони. Проте проведені пізніше кількісні розрахунки, що враховують реальні значення мас, коефіцієнтів пружності і в'язкості складових частин слухової системи, показали неспроможність цієї гіпотези. Інакше кажучи, спроба перетворити чисто якісну гіпотезу на точну модель, яку можна досліджувати математичними методами, зразу ж знайшла неспроможність початкових принципів. Звичайно, якщо ми побудували деяку модель і навіть отримали добру згоду між її властивостями і результатами відповідного біологічного експерименту, то це ще не доводить правильності моделі. Але якщо на підставі вивчення моделі можна зробити якісь прогнози про поведінку тієї біологічної системи, яку ми моделюємо, а потім підтвердити ці прогнози реальним експериментом, то це служить набагато вагомішим свідченням на користь правильності моделі.

Розглянемо деякі моделі, що відносяться до різних біологічних процесів і систем, вибираючи в першу чергу такі приклади, де можливе кількісне порівняння результатів дослідження моделі із експериментом.

1.2. Аксиоматичні моделі активних середовищ

1.2.1. Аксиоматика моделі.

Одна з характерних рис нервової тканини, є здібність до збудження і передачі збудження від одних ділянок клітини до інших, та вздовж мережі.

Збудження в нервовій клітині може виникнути або внутрішньо, або під впливом збудженої сусідньої клітини, або під дією зовнішнього сигналу, наприклад роздратуванням електричним імпульсом. Перейшовши в збуджений стан, клітина залишається в цьому стані якийсь час, а потім збудження зникає і настає так званий рефрактерний період, протягом якого клітина не реагує на поступаючі до неї сигнали. Потім клітина повертається в первинний стан спокою, із якого вона знову може перейти в збуджений стан.

Процес виникнення і поширення збудження має ряд чітко виражених властивостей, базуючись на яких можна побудувати формальну модель явища. Для дослідження такої моделі можна застосовувати математичні методи.

Модель можна досліджувати або аналітичними методами, або шляхом реалізації її на обчислювальній системі. Множина X (збудливе середовище) складається з кінцевого числа елементів. Для аналітичного дослідження можна припускати X деяким безпервною множиною (наприклад, вважати, що X — це площина). Простий варіант моделі отримаємо, прийнявши за X відрізок прямої (аналог нервового волокна) і припустивши, що час t , протягом якого кожен елемент знаходиться у збудженому стані дуже мал. Тоді процес послідовного поширення імпульсів по такому «нервовому волокну» легко описати ланцюгом звичайних диференціальних рівнянь першого порядку.

У складніших випадках (багатовимірне середовище, наявність розгалужень, облік тонших властивостей збудливої тканини) математичний опис процесу стає набагато важчою задачею. Описана модель грубій формі

відтворює основні властивості реальних збудливих тканин, вона здатна імітувати низку важливих явищ, пов'язаних з функціонуванням живого організму [1, 2].

1.2.2. Динамічна пам'ять.

Розглянемо двовимірне збудливе середовище, що володіє спонтанною активністю. Нехай всі елементи цього середовища мають один і той же період спонтанної активності T і одну і ту ж початкову фазу. Якщо таке середовище ізольоване від зовнішніх дій, то всі її елементи перейдуть в збуджений стан одночасно, потім збудження зникне, після чого через час T всі елементи збудяться знову і т.д. Припустимо, що в якийсь момент, коли всі елементи середовища знаходяться у спокої, порушимо один з них (позначимо його символом A) зовнішньою дією. Збудження почне розповсюджуватися від цього елемента по середовищу. В результаті хвиля збудження пробіжить по всіх елементах середовища і затухне, або в певний момент наступить спонтанне збудження тих елементів, до яких не встигла дійти хвиля від елемента A .

Через час T після того, як елемент A був збуджений зовнішнім стимулом, він збудиться знову. Потім почне збуджуватися решта елементів. Цього разу елемент A вже не буде джерелом поширення збудження; кожний з решти елементів збудиться спонтанно після часу T після попереднього його збудження. Послідовне збудження клітин повторює збудження коли воно було викликане стимуляцією клітини A . Через час $2T$ після первинної стимуляції знову почнеться поширення збудження від елемента A і т. д., тобто весь процес повторюватиметься періодично із періодом T .

Якщо порушити не один елемент, а деяку певну групу елементів в цьому випадку виникає періодичний процес поширення збудження. Таким чином, активне середовище може виконувати роль «динамічної пам'яті»: вона «запам'ятовує» завдане роздратування, його фазу і місце нанесення.

Можливо, що в реальних фізіологічних механізмах пам'яті, особливо якщо йдеться про «короткочасну» пам'ять, процеси, подібні тільки що описаному, виконують певну роль [2].

1.2.3. Відносна рефрактерність

Перехід елементів збудливого середовища із стану рефрактерності в стан спокою відбувається стрибком. Але в реальних нервових клітинах цей перехід відбувається поступово, тому між станом повної незбудливості - абсолютною рефрактерністю і станом спокою лежить стан відносної рефрактерності, в якому клітина відповідає на стимуляцію, але сила роздратування, необхідного для отримання відповіді, вище, ніж в нормі. Точніше кажучи, кожна збудлива клітина відповідає лише на роздратування, інтенсивність яких перевищує певну величину, звану порогом. У разі роздратування клітини електричним струмом повинна бути досягнута деяка мінімальна напруга. Виникення абсолютної рефрактерності після збудження можна трактувати як підвищення цього порогу до рівня, недосяжног без пошкодження клітини. Тоді відносна рефрактерність — це фаза, в якій відбувається поступове зниження порогу від значення, що відповідає абсолютній рефрактерності, до значення, відповідного стану спокою [3].

1.2.4. Формальний опис руху точкових імпульсів по волокну

У простих моделях процес поширення імпульсу можна описати математично. Розглянемо відрізок по якому розповсюджується збудження. Вважатимемо імпульси короткими коли у кожен момент часу збудження зосереджене в одній точці. Припустимо, що по волокну проходить серія імпульсів, $t_n(x)$ — момент часу, коли n -імпульс проходить через точку x , $R(x)$ — період рефрактерності волокна в точці x_s , тобто час, за який ця точка після збудження повертається в нормальний стан спокою. Волокно називається

однорідним, якщо $R(x) = \text{const}$, інакше його називають неоднорідним. Назвемо величину

$$\mu_n(x) = \min \left\{ \frac{t_n(x) - t_{n+1}(x)}{R(x)}, 1 \right\} \quad (1.1)$$

фазою рефрактерності волокна в точці x у момент проходження n -го імпульсу. Відповідно, природно вважати що швидкість поширення імпульсу визначається фазою рефрактерності: $v_n = v(j_n(x))$

При зниженні швидкості до деякого критичного рівня поширення збудження припиняється. Ця умова рівносильна припущенню про наявність фази абсолютної рефрактерності.

Тоді, вважаючи, що нервово волокно стимулюється (у початковій точці) періодично із періодом T , одержуємо, що проміжок часу між проходженнями $(n-1)$ -го і n -го імпульсів через точку x

$$t_n(x) - t_{n-1}(x) = T + \int_0^x \left[\frac{1}{v(\varphi_n(x))} - \frac{1}{v(\varphi_{n-1}(x))} \right] d\xi. \quad (1.2)$$

З (1.1) і (1.2) отримаємо диференціальне рівняння для $j_n(x)$:

$$\varphi_n'(x) = \frac{1}{R(x)} \left[\frac{1}{v(\varphi_n(x))} - \frac{1}{v(\varphi_{n-1}(x))} - \frac{dR(x)}{dx} \varphi_n(x) \right]. \quad (1.3)$$

Функція $j_n(x)$ задовольняє початковій умові

$$\varphi_n(0) = \frac{T}{R(0)}. \quad (1.4)$$

Підставляємо в (1.3) конкретний вид функції $v(j(x))$ та отримаємо рівняння, що описує процес поширення імпульсів по волокну.

Дійсно, якщо початкова фаза рефрактерності в кожній точці (тобто функція $j_0(x)$ відома, то із рівняння (1.3) і початкової умови (1.4) знаходяться послідовно $j_1(x)$, $j_2(x)$ і т. д., а по них, у свою чергу, із рівності (2) визначаються $t_1(x)$, $t_2(x)$ і т.д. В результаті для кожної точки x знаємо час проходження через неї кожного імпульсу, тобто отримуємо повний опис процесу.

Модельні дослідження зручні тим, що дозволяють детально досліджувати роль параметрів [4-6].

1.3. Іонно-мембранні механізми збудження

Механізм поширення збудження

Природодослідник Дж. Прістлі (1733—1804) виказав припущення, що передача збудження по нерву є рухом «електричної рідини». У 1791 році Л. Гальвані розвинув вчення про «тваринну електрику».

Е. Дюбуа-Реймон удосконалив техніку, і що одержав надійні кількісні дані про електричні потенціали в нервових і м'язових тканинах у спокої і при збудженні.

Умрат М. (у 1928 р.) запропонував диференціальні рівняння для опису процесу збудження, вважаючи, що поширення збудження пов'язане з дифузією хімічно активної речовини.

На основі досліджень Б.Ф. Веріго, Ю. Бернштейна, Л. Германа і інших, виявлено, що збуджена ділянка волокна стає генератором електричного струму [7]

Нервові і м'язові волокна по своїй електричній природі подібні провіднику, тобто є провідний сердечник (протоплазма), оточений оболонкою (мембрана та волокна) із великим опором і ємкістю. Основна роль у формуванні і розповсюдженні імпульсу належить мембрані. Якщо із нервового волокна видавити протоплазму і заповнити його оболонку (мембрану) морською водою, то таке нервове волокно може протягом досить довгого часу передавати збудження. Ділянка волокна збуджується, коли

різниця потенціалів по обидві сторони мембрани досягає деякого порогового рівня. Різниця потенціалів на мембрані служить параметром, що визначає включення механізмів генерації струму.

В стані спокою між зовнішньою поверхнею нервового волокна і протоплазмою, що знаходиться всередині, існує різниця потенціалів, що становить приблизно 60—90 мілівольт, причому поверхня клітини заряджена позитивно по відношенню до протоплазми. Ця різниця потенціалів називається потенціалом спокою. Вона створюється тим, що іонний склад протоплазми нервових і м'язових клітин сильно відрізняється від іонного складу навколишньої позаклітинної рідини; у протоплазмі концентрація іонів калія в 30—50 разів більше, а іонів натрію — в 10 разів менше, ніж зовні. Позитивно заряджені іони калія, виходячи через клітинну мембрану назовні, створюють потенціал спокою. Іони натрію, проходячи у зворотному напрямі, створюють протилежний ефект, але оскільки в стані спокою мембрана значно більш проникна для калія, ніж для натрію, перший процес домінує, і в результаті виникає різниця потенціалів [8-11].

Коли мембрана переходить в збуджений стан (під впливом зовнішньої дії, збудження сусідніх ділянок мембрани і т. д.) виборчим чином міняється її проникність для різних іонів. Спершу відбувається різке збільшення натрієвої проникності мембрани і потік позитивно заряджених іонів натрію спрямовується всередину клітини. При цьому різниця потенціалів по обидві сторони мембрани спершу знижується до нуля, а потім міняє знак: аксоплазма стає зарядженою позитивно по відношенню до позаклітинного середовища, причому різниця потенціалів досягає в максимумі приблизно 30 мв. (Таким чином, повна зміна мембранного потенціалу в порівнянні з початковим значенням складає приблизно 100—120 мв.) Після цього знову починає переважати потік іонів калія назовні, і система поступово повертається до початкового стану спокою.

Тривалість процесу збудження залежить від температури. Це означає участь в роботі мембранного генератора хімічних процесів.

Знайдемо розподіл потенціалу V уздовж волокна, вважаючи його пасивним, тобто повністю позбавленим здатності до генерації імпульсів. Для простоти розглядаємо випадок, коли потенціал V_0 , створений зовнішнім струмом, підтримується тривалий час, всі перехідні процеси, обумовлені ємкістю мембрани, встигають закінчитися.

Нехай i —подовжній струм, i_m — струм через мембрану, що доводиться на одиницю довжини волокна, r_i і r_m — відповідно опори протоплазми і мембрани на одиницю довжини волокна. Якщо D_i — зміна струму i на ділянці D_x , то $D_i/D_x = i_m$,

Згідно закону Ома,

$$i = \frac{\Delta V}{r_i \Delta x}$$

Підставляємо значення для i і i_m в рівність

$$i_m = \frac{\Delta i}{\Delta x}$$

Переходимо до ліміту при $D_x > 0$, отримаємо, що розподіл потенціалу уздовж (пасивного) волокна задається диференціальним рівнянням другого порядку

$$\frac{r_m}{r_i} \frac{d^2 V}{dx^2} = V$$

Його рішення (з урахуванням того, що потенціал не може необмежено рости уздовж волокна) має вигляд

$$V = V_0 e^{-x/l}$$

де $l = (r_m / r_i)^{1/2}$ - константа довжини волокна, тобто довжина ділянки, на якому потенціал зменшується в e разів. Для реального нервового волокна l дорівнює 1-2 міліметрів.

Поширення збудження по волокну залежить від електричних параметрів волокна і навколишнього середовища та від властивостей мембрани як генератора імпульсів. У відповідному образом поставленому експерименті ці два аспекти можна вивчати певною мірою незалежно [12]

1.4. Моделі поширення збудження

Поширення електричного імпульсу по кабелю, що містить генератори струму, описується рівнянням вигляду

$$\frac{a}{2R} \frac{\partial^2 V}{\partial x^2} = C \frac{\partial V}{\partial x} + I(V).$$

Де $I(V)$ характеризує властивості генераторів струму, R — питомий опір провідника, C — питома місткість оболонки, a — діаметр провідника. Спрощене рівняння, що описує поширення імпульсу по нервовому волокну, одержимо, прийнявши за $I(V)$ деяку фіксовану функцію змінних x і t .

Одержане таким чином рівняння досить добре відтворює деякі властивості реального процесу поширення імпульсу [13]. Проте для точнішого опису цього процесу потрібна складніша модель, що враховує основні властивості мембранного генератора. Така модель була в 1952 році запропонована фізіологами А. Ходжкином і А. Хакслі [10].

У своїх теоретичних побудовах А. Ходжкін і Д. Хакслі виходили з результатів експериментального дослідження гігантського аксона кальмара.

У моделі Ходжкіна—Хакслі $I(V)$ представляється як сума трьох елементів — калієвого струму, натрієвого струму і струму, створюваного

іншими переносниками зарядів (зокрема, іонами хлора). Моделлю Ходжкина—Хакслі називається наступна система нелінійних диференціальних рівнянь:

$$\begin{aligned}
 C \frac{dV}{dt} &= g_K (V - V_K) + g_{Na} (V - V_{Na}) + I(t), \\
 g_K &= g_{K \max} * n^4, \\
 g_{Na} &= g_{Na \max} * m^3 * h, \\
 \frac{dn}{dt} &= \alpha_n (1 - n) - \beta_n * n, \\
 \frac{dm}{dt} &= \alpha_m (1 - m) - \beta_m * m, \\
 \frac{dh}{dt} &= \alpha_h (1 - h) - \beta_h * h, \\
 \alpha_n &= \frac{0,01(V - 10)}{1 - e^{(10-V)/10}}, \\
 \alpha_m &= \frac{0,1(V - 25)}{1 - e^{(25-V)/10}}, \\
 \alpha_h &= 0,7e^{-V/20}, \\
 \beta_n &= 0,125e^{-V/80}, \\
 \beta_m &= 4e^{-V/18}, \\
 \beta_h &= \frac{1}{1 + e^{(30-V)/10}}.
 \end{aligned} \tag{1.5}$$

Система рівнянь Ходжкина—Хакслі не може бути розв'язана аналітично в явному вигляді. Рішення цієї системи, що відповідає певним початковим і краєвим умовам, можна одержати за допомогою чисельних методів. Деякі властивості процесу поширення імпульсу по нервовому волокну можна витягнути із аналізу системи (5) не маючи її рішення. Роблячи допущення, можна замінити систему (1.5) її спрощеним наближенням і таким чином полегшити задачу дослідження моделі.

Припустимо, що нервеве волокно, що знаходилося первинно у спокої, стимулюється впродовж короткого часу зовнішнім джерелом струму. Як показує аналіз рівнянь, при цьому перш за все почне зростати величина $m(V, t)$, що приведе до збільшення натрієвого струму

рівного $b_n = 0.125 \exp(V / 80)$, а отже і до збільшення різниці потенціалів між протоплазмою і зовнішнім середовищем (тобто посиленню деполяризації волокна). При збільшенні деполяризації величина m росте, а величина h зменшується, причому ці зміни m і h зв'язані так, що величина m^3h , що входить у вираз для натрієвого струму, спершу росте, а потім починає убувати.

Функція $n(V, t)$ під дією зовнішнього джерела струму теж починає зростати, але із істотним запізнюванням в порівнянні з m . Зростання n означає посилення калієвого струму. Оскільки калієвий струм має напрям, протилежний напрямку натрієвого струму, його збільшення зменшує деполяризацію, повертаючи кінець кінцем мембранний потенціал до початкового рівня.

Якщо, підставивши в рівняння (1.5) значення констант, одержані із експериментів на реальному аксоні, знайти шляхом чисельних розрахунків форму імпульсу, що виникає під дією зовнішнього стимулу, то ми побачимо, що модель Ходжкина—Хакслі не тільки якісно, але і кількісно узгоджується із даними фізіологічного експерименту.

Зміна потенціалу мембрани призводить до збудження нервового волокна і виникнення в ньому імпульсу у випадку, коли ця початкова зміна перевищує деяку певну величину, звану порогом.

Наявність у всіх нервових і м'язових волокон рефрактерності, тобто деякого періоду незбудливості, наступаючого після проходження імпульсу, теж природно витікає із даної моделі.

З експерименту відомо, що якщо діяти на нервові волокна струмом поступово наростаючої сили, то така дія призводить до виникнення імпульсу у волокну лише в тому випадку, якщо швидкість наростання струму перевищує деяку певну величину; при дуже повільному збільшенні струму волокно не збуджується (це явище називається «акомодацією», або «звиканням» волокна). У моделі Ходжкина—Хакслі наявність акомодації пояснюється тим, що при повільному наростанні струму встигає розвинутиися натрієва інактивация (тобто зменшиться повільна змінна A), через що натрієвий струм

не може перевищити калієвий і імпульс не виникає. Мінімальну швидкість наростання струму, при якій ще можливе виникнення імпульсу, можна визначити за допомогою чисельного рішення рівнянь Ходжкина — Хакслі.

Рішення системи (1.5) дозволяє одержати характеристики імпульсу, що розповсюджується по волокну. Як було показано в 1952 р. Ходжкином і Хакслі, їх модель дає правильні значення для швидкості поширення імпульсу і добре відтворює його форму. Зокрема, амплітуда потенціалу дії, що розповсюджується, декілька нижчі, ніж амплітуда імпульсу, що виникає при одночасному збудженні всієї мембрани. Модель дозволила також визначити залежність швидкості поширення імпульсу від параметрів волокна і умов зовнішнього середовища (температури, питомого опору).

Модель Ходжкина—Хакслі дозволила розрахувати зміни форми другого імпульсу, що виникає через різні терміни після першого. Ця модель дозволила обчислити зміну опору мембрани в різні фази імпульсу, відповіді мембрани на роздратування постійним струмом і т.д. Розрахункові дані добре співпадали із результатами відповідних експериментів.

Отже, модель Ходжкина — Хакслі достатньо добре відтворює, притому не тільки якісно, але і кількісно, основні особливості процесу збудження реального нервового волокна. Разом із тим при створенні цієї моделі такі властивості волокна як рефрактерність, акомодация, наявність порогу і т.д. зовсім не вводилися в модель в явному вигляді. Ця модель була побудована на підставі досить обмеженого набору експериментальних даних, одержаних на аксоні кальмара в умовах так званої «фіксації напруг»: спеціальна електронна апаратура доводить різницю потенціалу по обидві сторони мембрани до заданої величини і підтримує її на цьому рівні, незалежно від величини струму, що генерується самою мембраною. При цьому вимірюється величина струму, що відповідає різним значенням різниці потенціалів.

Рівняння Ходжкина—Хакслі можуть бути застосовані для дослідження процесів, що відбуваються в збудливих середовищах із різною геометрією (волокна, що гілкуються, волокна змінного діаметру і т. д.).

1.5. Обчислювальні можливості нейронів

Цікаві результати були одержані при аналізі процесів, що відбуваються в місцях галуження нервового волокна. Виявилось, що вузол галуження може виконувати роль логічного елемента, що реалізовує основні логічні операції над сигналами. Залежно від діаметрів трьох гілок, що сходяться у вузлі, із яких по двох сигнал поступає до вузла галуження, а по одній виходить із нього, можливі наступні випадки:

а) надходження імпульсу хоча б по одному з входів породжує імпульс на виході; інакше кажучи, вузол здійснює логічну функцію «або»;

б) імпульс на виході виникає в тому і лише тому випадку, коли до вузла приходять одночасно два імпульси, по обох вхідних гілках; це — логічна функція «і»;

с) імпульс, що приходить по першому входу, не порушує вихідну гілку, а що приходить по другому входу — порушує її. Проте, якщо спочатку приходить імпульс по першій вхідній гілці, а через деякий час — по другій, то цей другий імпульс теж не порушить вихідну гілку. Отже, збудження на виході виникає в тому і лише в тому випадку, якщо другий вхід збуджений, а перший немає, тобто імпульс, що приходить по першій гілці, виконує роль блокування (Причина тут полягає у тому, що імпульс, що приходить по першій (достатньо тонкої) гілці, не може порушити вузол, але викликає там деяку натрієву інактивацию, що призводить до підвищення порогу збудливості у вузлі.).

Як відомо із математичної логіки, комбінуючи вказані вище операції («и», «або», блокування) можна одержати будь-які логічні функції. Інакше кажучи, в нервових структурах (таких, як, наприклад, дендрити клітин), що гілкуються, може відбуватися вельми складна обробка інформації.

1.6. Висновки до першого розділу

У розділі було продемонстровано ключову роль математичних моделей в біологічних дослідженнях та визначено зворотній зв'язок, коли біологічні моделі стають прикладом для побудови математичних об'єктів та моделей.

Представлено аксіоматичні моделі активних середовищ, а саме аксіоматика моделі нейрону, прояв функціонування моделей у вигляді реалізації динамічної пам'яті. Розглянуто механізм , що має назву - відносна рефрактерність.

Розглянуто іонно-мембранні механізми збудження, на яких базуються сучасні теорії нервової клітини.

Розглянуто моделі поширення збудження, яке і реалізує перетворення напруги у керуючий сигнал.

Наведено ідеї, що до обчислювальних можливостей нейронів.

РОЗДІЛ 2. КЛАСИЧНІ ОБЧИСЛЮВАЛЬНІ АРХІТЕКТУРИ

2.1. Архітектура фон Неймана

Архітектура фон Неймана – це модель організації комп'ютерної системи, запропонована американським математиком фон Нейманом у сорокових роках 20-ого століття. Вона описує ключові елементи, з яких повинен складатися комп'ютер, і як ці елементи повинні взаємодіяти між собою. Більшість сучасних комп'ютерів тією чи тією мірою слідує цій архітектурі. Ця архітектура виходить із того, що комп'ютер має процесор та оперативну пам'ять, з'єднані між собою шиною. Пам'ять використовується як для зберігання інструкцій команд, так і для зберігання даних. Програма є набором команд, які повинен виконати процесор. Він робить це в циклі: бере з пам'яті команду та виконує її, потім бере наступну команду та виконує і так далі. Крім процесора та оперативної пам'яті, які є ядром будь-якої комп'ютерної системи, до неї також входять пристрої для введення та виведення інформації, їх називають периферійними.

Оперативна пам'ять є масив осередків у кожному у тому числі зберігається якесь бітове значення. Кожен осередок має свою унікальну адресу. Оперативна пам'ять зберігає як інструкції, так і дані. Процесор може звернутися до будь-якої комірки, знаючи її адресу.

Частина комп'ютера, яка відповідає за виконання команд, називається процесором. Процесор складається з трьох ключових елементів: арифметико-логічного пристрою, який, як можна зрозуміти з назви, виконує логічні та арифметичні операції над даними (наприклад, додавання або інверсія), керуючого пристрою, який є "диригентом", що координує роботу всіх інших пристроїв і регістрів, які тимчасово зберігають у собі інформацію, необхідну процесору зараз. Регістри бувають спеціального призначення та загального призначення.

Регістри загального призначення це свого роду "руки", в які процесор бере дані з якими зараз працює. Такі регістри можуть бути як введення чи виведення для арифметико-логічного устрою. Щоб, наприклад, скласти два числа, які зберігаються в оперативній пам'яті, керуючий пристрій повинен звернутися до оперативної пам'яті і перенести дані звідти в регістри загального призначення, повідомити арифметично-логічний пристрій, в яких регістрах зберігаються значення чисел, повідомити арифметично-логічний пристрій, що необхідно виконати операцію додавання і також повідомити в який регістр необхідно вивести результат цієї операції. Щоб мати можливість взаємодіяти один з одним, процесор і оперативна пам'ять з'єднані один з одним набором проводів, званих шиною. За допомогою шини процесор може читати дані з оперативної пам'яті, повідомляючи їй адресу комірки, що цікавить, і посылаючи запит на читання. Подібним чином процесор може записувати дані в оперативну пам'ять, передаючи їй адресу комірки, в яку необхідно зробити запис, і надсилаючи запит на запис.

Як уже було зазначено, комп'ютер потребує периферійних пристроїв для введення та виведення інформації. Взаємодія процесора з цими пристроями багато в чому схожа на взаємодію процесора з оперативною пам'яттю. Периферійні пристрої також підключені до шини і їм також можна надсилати та приймати дані за допомогою спеціальних команд. Кожен пристрій має свою адресу та активується, якщо ця адреса з'являється на шині. Після цього з ним можна працювати – приймати чи надсилати інформацію.

2.2. DSP-процесори

Існують різні комп'ютерні системи: з процесорами загального призначення (general-purpose, в основному x86) у ноутбуках та робочих станціях, їх потужними багатоядерними версіями в датацентрах, мобільними процесорами у телефонах, цифровими контролерами у побутовій техніці та на транспорті. Але крім всіх згаданих варіантів є ще одне важливе сімейство:

цифрові сигнальні процесори, частіше звані Digital Signal Processors або просто DSP.

Саме DSP вирішують завдання обробки великих обсягів інформації в реальному часі, що виникають при передачі даних у мобільних мережах, обробці фотографій та відновленню звуку. Навіть у топових телефонах вся ця робота виконується не на потужних ARM-ядрах, а на спеціалізованих DSP.

Перші DSP з'явилися у 1970-х роках. Ці процесори стали логічним розвитком спеціалізованих аналогово-цифрових пристроїв, призначених для обробки мовлення, насамперед її кодування та фільтрації (прорив у відповідних науково-технічних галузях став можливим завдяки попиту на ці технології у роки Другої Світової війни). Трудомісткість і складність розробки пристроїв під кожну задачу, а також успіхи у розвитку електронної бази (широке поширення технології MOSFET) та математичних алгоритмів (БПФ, цифрова фільтрація) призвели до можливості створення універсальних, тобто програмованих, цифрових процесорів, які могли бути за допомогою програм адаптовані для широкого класу задач. практиці означала зниження вартості розробок, скорочення часу виходу ринку (time-to-market), можливість післяпродажного оновлення алгоритму для усунення помилок, можливість підтримки нових вимог користувачів.

Через необхідність обробки реального часу та економії електроенергії DSP сильно відрізнялися від процесорів загального призначення. Вони були першим прикладом програмованих обчислювальних прискорювачів, тобто процесорів, які максимально ефективно вирішують певний клас завдань.

Відрізняються DSP від звичайних потужних процесорів загального призначення,

Значна частина площі кристала Pentium відводиться не під обчислювальні ресурси, а під складну логіку визначення залежностей, спекулятивного виконання (out-of-order speculative execution) та складання розкладу (scheduling). У сумі накладні витрати призводять до того що, що

“ККД” процесора, тобто. енергія, що витрачається виконання реальних обчислень, становить менше 1% :

У той час як проста арифметична операція вимагає близько 0,5–20 пДж, сучасні ядра витрачають близько 2000 пДж на її планування.

Звичайні багатоядерні процесори споживають у 157–707 разів більше енергії, ніж апаратне забезпечення, створене на замовлення зі статті "Rise and Fall of Dark Silicon").

Таблиця 2.1.

Порівняльні характеристики Skylake Xeon Platinum 8180M та TMS320C6713BZDP300

Параметр	ЦП (Intel)	DSP (IT)
Частота	2.5 ГГц	500 МГц
Число ядер	28	1
Пікова продуктивність	560 ГПС	1.8 GIPS
Енергоспоживання	205 Вт	1 Вт
Вийшла з ладу	так	Ні
Ціна	\$1300 (+ система охолодження)	\$35
Цільові програми	Будь-які	
Продуктивність	0.097 GIPS/Вт/ядро	1.7 GIPS/Вт/ядро (у 17 разів краще)
Продуктивність	0,0075 IPS/Вт/ядро/\$	0.051 GIPS/Вт/ядро/\$ (У 7000 разів краще)

Абсолютні значення продуктивності у DSP істотно нижчі, питома продуктивність на 1 ват споживаної потужності вище на 4 порядку. БПФ, фільтрації та ін. Такі завдання характеризуються високою регулярністю оброблюваних даних (плоскі масиви в пам'яті, оброблювані в циклах) і більшим паралелізмом за даними.

Специфіка розв'язуваних завдань істотно впливає на архітектуру та системне ПЗ для DSP. За словами Jennifer Eyre, аналітика дослідницького центру BDTI, "архітектура DSP формується тими завданнями, які на них вважаються" ("Evolution of DSP Processors"). Перерахуємо особливості таких завдань:

- Практично нескінченний паралелізм рівня команд (ILP, Instruction Level Parallelism)

- Більшість алгоритмів (згортка, швидке перетворення Фур'є, обчислення з комплексними *числами*) зводяться до виконання операцій складання та множення над щільними масивами даних

- Обчислення проводяться на вбудованих системах, з жорсткими вимогами щодо енергоспоживання

Таким чином, основними цілями є максимальне використання властивого задач паралелізму і зниження енергоспоживання при виконанні циклів.

Для використання ILP використовуються різні техніки:

Векторні інструкції (SIMD, Single Instruction Multiple Data)

Складні інструкції (CISC, Complex Instruction Set Computer):

1. Складові математичні операції (множення та віднімання з накопиченням, гістограми, спецфункції, комплексні *обчислення*)
2. Операції множення зі зрушенням (для арифметики з фіксованою точкою)
3. Широкий набір режимів адресації (з кроком, з пре-і пост-інкрементом, циклічним обходом та ін.)
4. Алгоритмо-специфічні операції (наприклад, підрахунок контрольних сум мережевих пакетів, криптографія, аудіо-відео декодери)
5. Розширювані системи команд (в IP-продуктах Ceva та Tensillica)
 - Сильно розширована пам'ять (для видачі кількох паралельних завантажень за такт або індексних звернень до пам'яті типу scatter/gather)
 - Звільнення від затримок, що викликаються розгалуженнями:

1. Підтримка передикатного виконання для всіх команд процесора (або переважної більшості)
2. Процесорні хінти (спеціальні інструкції для завантаження даних)
3. Слоти затримки
4. Швидкі цикли (zero-overhead loops)
 - Винесення найбільш обчислювально-ємних алгоритмів (алгоритм Вітербі, БПФ, QR-розкладання, нейромережових обчислень) на вбудовані прискорювачі (т.зв. fixed function units)
 - Прискорення тривалих операцій з пам'яттю за допомогою спеціальних блоків прямого доступу до пам'яті (DMA), за допомогою довільних 2D/3D-кроків

Для зниження ж енергоспоживання використовуються

- Перехід від дійсних чисел на арифметику з фіксованою точкою
- Нестандартні типи даних (наприклад 20- та 40-бітові цілі)
- Упорядковані (in-order) обчислення, відсутність спекулятивності (speculation) та позачергового виконання (out-of-order)
 - Явне формування паралельно виконуваних пакетів інструкцій компілятором (VLIW)
 - Відсутність відстеження процесором залежностей між інструкціями та перенесення відповідальності за точне планування інструкцій на компілятор (exposed pipeline)

У процесорах загального призначення кожне із зазначених рішень призвело б до фатального погіршення продуктивності та зручності роботи, але у разі DSP особливості застосування знімають цю проблему.

Компілятор є невід'ємним та найважливішим компонентом DSP як продукту. У разі процесорів загального призначення різниця між поганим і хорошим компілятором зазвичай невелика т.я. процесор компенсує неефективність за допомогою out-of-order виконання. У той же час для DSP цей розкид легко досягає десятка. Крім того, складність програмування на

асемблері набагато вища, тому переписування на ньому всього performance-critical коду зазвичай неможливе.

Компілятори DSP зазвичай засновані на open-source рішеннях:

- Open64 (ранні продукти Ceva та Cadence/Tensilica)
- GCC (Texas Instruments та ранні продукти Qualcomm)
- LLVM (нові продукти Ceva, TI та Qualcomm, можливо Cadence/Tensilica)

В основному це пов'язано зі складністю самостійної підтримки нових версій C++, що підходить для комерційного використання ліцензією та підтримкою нових мов, таких як OpenCL, OpenMP/OpenACC та Halide.

2.3. Нейронні мережі.

Нейронна мережа – це метод у штучному інтелекті, який вчить комп'ютери обробляти дані у такий самий спосіб, як і людський мозок. Це тип процесу машинного навчання, званий глибоким навчанням, який використовує взаємопов'язані вузли або нейрони у шаруватій структурі, що нагадує людський мозок. Він створює адаптивну систему, за допомогою якої комп'ютери навчаються. Штучні нейронні мережі намагаються вирішувати складні завдання.

Нейронні мережі допомагають комп'ютерам приймати розумні рішення з обмеженою участю людини. Вони можуть вивчати та моделювати відносини між нелінійними та складними вхідними та вихідними даними.

Архітектура нейронних мереж повторює структуру людського мозку. Клітини людського мозку, звані нейронами, утворюють складну мережу з високим ступенем взаємозв'язку та посиляють один одному електричні сигнали, допомагаючи людям обробляти інформацію. Так само штучна нейронна мережа складається з штучних нейронів, які взаємодіють на вирішення проблем. Штучні нейрони - це програмні модулі, які називаються вузлами, а штучні нейронні мережі - це програми або алгоритми, які

використовують обчислювальні системи для виконання математичних обчислень.

Базова нейронна мережа містить три шари взаємопов'язаних штучних нейронів:

Вхідний шар

Інформація із зовнішнього світу надходить у штучну нейронну мережу із вхідного шару. Вхідні вузли обробляють дані, аналізують або класифікують їх та передають на наступний шар.

Прихований шар

Приховані шари одержують вхідні дані від вхідного шару або інших прихованих шарів. Штучні нейронні мережі можуть мати велику кількість прихованих шарів. Кожен прихований шар аналізує вихідні дані попереднього шару, обробляє їх та передає на наступний шар.

Вихідний шар

Вихідний шар дає остаточний результат обробки всіх даних штучною нейронною мережею. Він може мати один чи кілька вузлів. Наприклад, під час вирішення завдання двійкової класифікації (так/ні) вихідний шар матиме один вихідний вузол, який дасть результат «1» чи «0». Однак у разі множинної класифікації вихідний шар може складатися більш ніж з одного вихідного вузла.

Архітектура глибокої нейронної мережі

Глибокі нейронні мережі або мережі глибокого навчання мають кілька прихованих шарів із мільйонами зв'язаних один з одним штучних нейронів. Число, зване вагою, свідчить про зв'язку одного вузла коїться з іншими. Вага є позитивним числом, якщо один вузол збуджує інший або негативним, якщо один вузол пригнічує інший. Вузли з вищими значеннями ваги мають більший вплив інші вузли.

Теоретично глибокі нейронні мережі можуть зіставляти будь-який тип введення будь-яким типом виведення. Однак їм потрібне набагато складніше навчання, ніж інші методи машинного навчання. Таким вузлам потрібні

мільйони прикладів навчальних даних, а не сотні чи тисячі, як у випадку із простими мережами.

Штучні нейронні мережі можна класифікувати тому, як дані передаються від вхідного вузла до вихідного вузла.

Нейронні мережі прямого розповсюдження

Нейронні мережі прямого поширення обробляють дані в одному напрямку від вхідного вузла до вихідного вузла. Кожен вузол одного шару пов'язаний із кожним вузлом наступного шару. Нейронні мережі прямого поширення використовують процес зворотний зв'язок поліпшення прогнозів.

Штучні нейронні мережі постійно навчаються, використовуючи цикли зворотного зв'язку, що коригують, для поліпшення своєї прогностичної аналітики. Дані протікають від вхідного вузла до вихідного вузла по безлічі різних шляхів нейронної мережі. Правильним є лише один шлях, який зіставляє вхідний вузол із правильним вихідним вузлом. Щоб знайти цей шлях, нейронна мережа використовує петлю зворотного зв'язку.

Кожен вузол робить припущення про наступний вузол на шляху.

Він перевіряє, чи є припущення правильним. Вузли надають більш високі значення ваги коліям, які призводять до більш правильних припущень, і нижчі значення ваги вузлів, які призводять до неправильних припущень.

Для наступної точки даних вузли роблять новий прогноз, використовуючи шляхи з вищою вагою, а потім повторюють крок.

Згорткові нейронні мережі

Приховані шари в нейронних мережах згортання виконують певні математичні функції (наприклад, підсумовування або фільтрацію), звані згортками. Вони дуже корисні для класифікації зображень, оскільки можуть витягувати з них відповідні ознаки, корисні для розпізнавання та класифікації. Нову форму легше обробляти без втрати функцій, які мають вирішальне значення для правильного припущення. Кожен прихований шар витягує та обробляє різні характеристики зображення.

Навчання нейронної мережі - це процес навчання нейронної мережі виконання завдання. Нейронні мережі навчаються шляхом первинної обробки кількох великих наборів розмічених чи нерозмічених даних. На основі цих прикладів мережі можуть більш точно опрацьовувати невідомі вхідні дані.

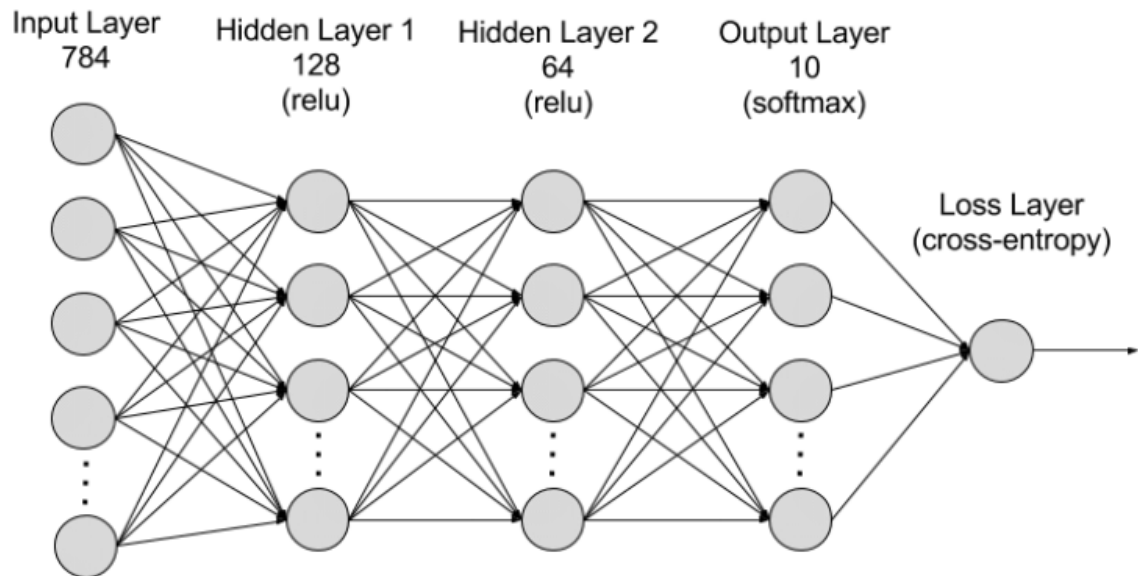


Рисунок 2.1. - Схема нейронної мережі. [https://d1.awsstatic.com/whatisimg/intro-gluon-1%20\(1\).ac2f31378926b5f99a4ba9d741c4aebe3b7a29e2.png](https://d1.awsstatic.com/whatisimg/intro-gluon-1%20(1).ac2f31378926b5f99a4ba9d741c4aebe3b7a29e2.png)

Традиційні методи машинного навчання вимагають участі людини, щоб програмне забезпечення працювало належним чином. Фахівець із роботи з даними вручну визначає набір відповідних функцій, які має аналізувати програмне забезпечення. Це обмеження робить створення та управління програмним забезпеченням трудомістким процесом.

При глибокому навчанні фахівець із роботи з даними надає програмному забезпеченню лише необроблені дані. Мережа глибокого навчання здобуває функції самостійно та навчається більш незалежно. Вона може аналізувати неструктуровані набори даних (наприклад, текстові документи), визначати пріоритети атрибутів даних та вирішувати складніші завдання.

Шари активації – це один з основних типів шарів, які використовуються у нейронних мережах. Вони є функцією, яка додає нелінійність до виходу

попереднього шару. Це дозволяє нейронні мережі краще моделювати складні функції і більш точно прогнозувати результати.

Шари активації приймають на вхід результати попереднього шару, які називають входом, і перетворюють їх у вихідне значення, яке передається наступному шару. Для цього вони використовують функцію активації, яка визначає, яким чином дані будуть перетворені.

Сигмоїдна функція активації - це нелінійна функція, яка перетворює вхідне значення в діапазоні від негативної нескінченності до позитивної нескінченності значення від 0 до 1. Ця функція активації часто використовується в нейронних мережах для завдань бінарної класифікації.

Математично сигмоїдна функція активації визначається:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Графічно сигмоїдна функція активації виглядає як S-подібна крива, що монотонно зростає і має асимптоти на 0 і 1. Зокрема, якщо $x > 0$, то $f(x) > 0.5$, а якщо $x < 0$, то $f(x) < 0.5$.

Сигмоїдна функція активації використовується перетворення вихідного значення нейрона на ймовірність, тобто. ймовірність того, що вхідне значення відноситься до класу 1, якщо ми працюємо із завданням бінарної класифікації. Якщо значення сигмоїдної функції близько 1, то ймовірність того, що вхідне значення відноситься до класу 1, висока. Якщо значення близько 0, то ймовірність того, що вхідне значення відноситься до класу 1, низька.

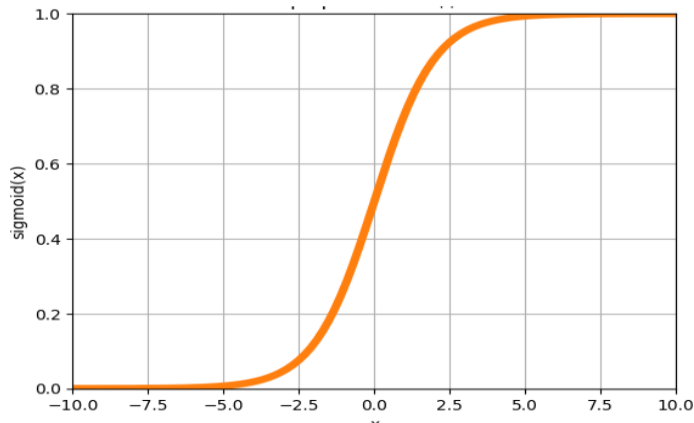


Рисунок 2.2. - Функція активації.

2.4. Висновки до другого розділу

Обчислювальні архітектури — це структури та принципи організації комп'ютерних систем для ефективного виконання обчислень. Існують різні типи архітектур, які визначають, як дані обробляються, зберігаються та передаються в межах комп'ютерних систем.

Архітектура фон Неймана. Принцип: Всі інструкції і дані зберігаються в пам'яті, і процесор по черзі виконує ці інструкції. Характеристики: Спільне використання пам'яті для зберігання як програм, так і даних. Використовує одну шину для доступу до пам'яті для даних і інструкцій. Приклад: Сучасні персональні комп'ютери, сервери.

Архітектура Гарварда. Принцип: Інструкції та дані зберігаються в різних блоках пам'яті, що дозволяє одночасно працювати з ними. Характеристики: Відокремлені шини для інструкцій та даних. Швидший доступ до пам'яті. Приклад: Вбудовані системи, цифрові сигнальні процесори (DSP).

Многопроцесорні архітектури. Принцип: Використовують кілька процесорів, що працюють паралельно для виконання обчислень. Типи: SMP (Symmetric MultiProcessing) — кілька однакових процесорів працюють над загальним обчислювальним завданням. NUMA (Non-Uniform Memory Access) — декілька процесорів мають власну локальну пам'ять, але можуть доступатися до пам'яті інших процесорів. Приклад: Сервери, суперкомп'ютери.

Архітектура квантових комп'ютерів. Принцип: Використання квантових бітів (кубітів), які можуть перебувати в кількох станах одночасно, для виконання обчислень. Характеристики: Можливість виконання певних типів обчислень значно швидше, ніж класичними комп'ютерами. У роботі не досліджувалась.

Нейромережі – концептуально інший підхід, до вирішення обчислювальних задач.

Кожна архітектура має свої переваги та обмеження і застосовується в залежності від типу обчислювальних задач, що потребують виконання.

РОЗДІЛ 3. МОДЕЛІ НЕЙРОНІВ

3.1. Еквівалентна схема мембрани гігантського аксона кальмара

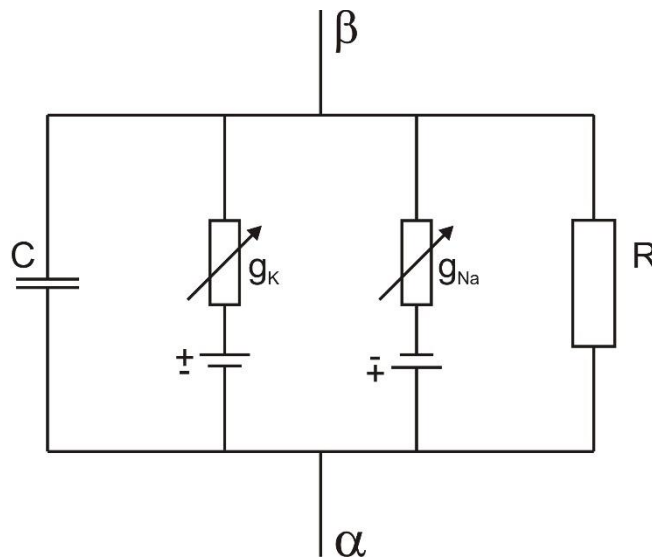


Рисунок 3.1. Еквівалентна схема мембрани гігантського аксона кальмара α – аксоплазма; β – моська вода, яка омиває аксон; C – конденсатор, який імітує мієлінову оболонку; g_{Na} , g_K – елементи-джерела постійного струму (натрієвий та калієвий), які мають протинаправлені е.д.с.; R – сумарний внутрішній опір натрія та калія.

У початковий момент часу елементи схеми знаходяться в стані рівноваги: потенціали g_K і g_{Na} врівноважують один одного, накопичення заряду на C не відбувається.

Як тільки на контактах α і β з'являється напруга, яка перевищує пороговий рівень, на клеммах конденсатора з'являється різниця потенціалів і відбувається накопичення заряду на C .

Заряд накопичується до тих пір, поки не буде досягнутий рівень напруги рівний різниці потенціалів між α і β . Після цього конденсатор починає розряджатися, змінюючи напрям струму в ланцюзі.

Розрядка відбувається до рівня меншого, ніж рівень в початковий момент часу. Після цього напрям струму в ланцюзі міняється знову і відбувається стабілізація схеми.

3.2. Спайкові моделі

Інтегратор і розряд

Одна з моделей в обчислювальній нейробиології, що найчастіше використовується - є квазіінтегруючий і розряджаючий (I&F) нейрон

$$v' = I + a - bv, \quad v \geq v_{thresh}, \text{ то } v \leftarrow c,$$

де v - «мембранний потенціал»; I - вхідний струм; a, b, c - параметри.

Коли мембранний потенціал v досягає граничного значення, то нейрон «розряджається» спайком, і v скидається в c

Такий нейрон є «збудливим класу 1»; він може розряджатися тонічними спайками з постійною частотою, і він є інтегратором.

Для реалізації ітерації $v(\tau + 1) = v(\tau) + (I + a - bv(\tau))$ потрібно всього 4 «операції із плаваючою точкою» (додавання, множення й т.д.), плюс одне порівняння з порогом.

Інтегратор і розряд, з адаптацією

«Модель I&F» - є одномірною, тому вона не може змодельювати залп і мати інших властивостей, властивим кортикальним нейронам. Введемо друге лінійного рівняння:

$$v' = I + a - bv + g(d - v)$$

$$g' = \frac{(e\delta(t) - g)}{\tau}$$

що описує динаміку активації високопорогового К-струму, тобто додати цієї моделі адаптацію частоти спайка. Дійсно, кожен розряд - збільшує К активаційний селектор g на дельта-функцію Дірака δ , і генерує вихідний струм, що сповільнює частоту тонічних спайків. Імітаційне моделювання по цій моделі - вимагає 10 «операцій із плаваючою точкою».

Інтегратор і розряд або залп

Сміт [24] запропонував для моделювання таламокортикальних нейронів модель - «інтегратор і розряд або залп» (I&FB):

$$v' = I + a - bv + gH(v - v_h)h(v_T - v)$$

$$\text{если } v = v_{thresh}, \text{ то } v \leftarrow c$$

$$h' = \begin{cases} \frac{-h}{\tau^-}, & \text{если } v > v_h \\ \frac{(1-h)}{\tau^+}, & \text{если } v < v_h \end{cases}$$

Де h - описує інактивацию кальцієвого Т-струму; $g, v_h, v_T, \tau^+, \tau^-$ - це параметри, що описують динаміку Т-струму; H - одинична східчаста функція (функція Хевисайда).

Одержуємо можливість моделювати залпи й інші режими.

Резонатор і розряд

Двомірним (2-D) аналогом «I&F нейрона» - є модель нейрона «резонатор і розряд» [16]:

$$z' = I + (b + i\omega)z$$

$$\text{если } \text{Im } z = a_{thresh}, \quad \text{то } z \leftarrow z_0(z)$$

де дійсна частина комплексної змінної z є «мембранним потенціалом». Тут b, ω, a_{thresh} - це параметри; $z_0(z)$ - довільна функція, що описує «залежний від активності» післяспайковий скид.

Квадратичний інтегратор

Альтернативою «квази I&F нейрону» - є «квадратичний I&F нейрон», відомий також за назвою «тета-нейрон» [2], [3] або «канонічна модель Ерментраута - Копелля»[10], виходячи з роботи [19]:

$$v' = I + a(v - v_{rest})(v - v_{thresh})$$

если $v = v_{peak}$, тогда $v \leftarrow v_{reset}$

де v_{rest} й v_{thresh} - це відповідно значення спокою й граничне значення «мембранного потенціалу». Цю модель варто вибрати для моделювання великомасштабних мереж інтеграторів.

Спайкова модель Іжикевича

$$v' + 0.004v^2 + 5v + 140 - u + 1$$

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

Змінна v - являє собою «мембранний потенціал» нейрона; u - це змінна відновлення мембрани, що враховує активацію K^+ іонних струмів й інактивацію Na^+ іонних струмів, і та, що забезпечує негативний зворотний зв'язок для v . Після того, як спайк досягає своєї вищої крапки (+30 мВ), - мембранна напруга й «змінна відновлення» - скидаються згідно (3). Якщо v більше 30: то тоді v спочатку скидається до 30, а потім – до c , щоб всі спайки були однакової амплітуди. Член $0.04v^2 + 5v + 140$ обраний так, щоб v був у мВ, а час був у мс.

Ця модель здатна демонструвати залпову поведінку всіх відомих типів кортикальних нейронів завдяки підбору параметрів. Для імітаційного моделювання 1 мс по цій моделі – потрібно 13 операцій із плаваючою точкою,

тому ця модель є досить ефективною для великомасштабних імітацій кортикальних мереж.

Модель Фіцхью – Нагумо

Параметри моделі Фіцхью – Нагумо

$$\begin{aligned}v' &= a + bv + cv^2 + dv^3 - u \\ u' &= \varepsilon(ev - u)\end{aligned}$$

можуть бути настроєні так, що ця модель описує спайкову динаміку багатьох нейронів-резонаторів.

Модель Хайдмарша – Роуза

Модель таламічного нейрона Хайдмарша - Роуза можна записати у вигляді:

$$\begin{aligned}v' &= u - F(v) + I - \omega \\ u' &= G(v) - u \\ \omega' &= \frac{(H(v) - \omega)}{\tau}\end{aligned}$$

де F, G, H – деякі функції. Залежно від вибору цих функцій, ця модель може, демонструвати всі розглянуті нейрокомп'ютерні властивості. Задача полягає у тому, як відшукати такі функції щоб змодельовати, наприклад, RS або LTS нейрони. Припустимо, що ці функції є поліномами третього ступеня. Так як нам потрібно імітувати форму «потенціалу дії», то максимальний часовий крок дорівнює 0.25 мс. Так як витрачається 30 «операцій із плаваючою точкою» / 0.25 мс часу імітаційного моделювання, то по цій моделі для моделювання 1 мс знадобиться 120 операцій.

Модель Морриса-Лекара

Моррис і Лекар запропонували просту 2-D модель для опису коливань у м'язовому волокні. Так як вона має параметри, що мають фізичний зміст і їх також можна виміряти, тому ця модель стала досить популярною в середовищі фахівців з обчислювальної нейробиології. Ця модель складається з рівняння «мембранного потенціалу» з миттєвою активацією «Са струму», і додаткового рівняння, що описує більш повільну активацію «К струму»:

$$\begin{aligned} C\dot{V} &= I - g_L(V - V_L) - g_{Ca}m_\infty(V) \\ &\quad \times (V - V_{Ca}) - g_Kn(V - V_K) \\ \dot{n} &= \lambda(V)(n_\infty(V) - n) \end{aligned}$$

Де

$$\begin{aligned} C\dot{V} &= I - g_L(V - V_L) - g_{Ca}m_\infty(V) \\ &\quad \times (V - V_{Ca}) - g_Kn(V - V_K) \\ \dot{n} &= \lambda(V)(n_\infty(V) - n) \end{aligned}$$

Ця модель може демонструвати різні типи спайків, - однак вона може демонструвати «тонічні залпи» тільки при додаванні додаткового рівняння, наприклад, повільної інактивації «Са струму». У цьому випадку, ця модель стає еквівалентної моделі Ходжкіна-Хакслі, (обидві ці моделі мають перехідні вхідні й незатухаючі вихідні струми).

У моделі Морриса-Лекара потрібно імітувати форму «потенціалу дії», т часовий крок повинен бути значно менше, ніж 1 мс. $\tau = 0.1 \text{ мс}$ - це найбільший крок, що дає розумні результати, якщо ця модель використовується для імітації кортикальних спайкових нейронів. Адже у цій моделі використовуються гіперболічні тангенси й експоненти, то для оцінки одного 0.1мс тимчасового кроку потрібно близько 60 «операцій із плаваючою точкою». Тобто 600 операцій/1 мс імітаційного моделювання.

3.3. Модель Ходжкіна-Хакслі

Модель Ходжкіна-Хакслі – це одна з найбільш важливих моделей в обчислювальній нейробиології. Вона складається з 4-х рівнянь і десятка параметрів, що описують мембранний потенціал, активацію «Na ланцюгів» й «K ланцюгів», інактивацію «Na ланцюга». Незважаючи на те, що ця модель демонструє досить обмежену поведінку при оригінальних значеннях параметрів, вона може демонструвати практично всі властивості, якщо відповідно підібрати її параметри.

У цілому, вчені відносять всі моделі, засновані на провідності, до типу Ходжкіна-Хакслі. Такі моделі важливі не тільки тому, що їхні параметри мають біофізичний сенс та є вимірюваними, але ще й тому, що ці моделі дозволяють досліджувати питання, пов'язані із синаптичним інтегруванням, дендритною кабельною фільтрацією, ефектами дендритної морфології, взаємодією між іонними струмами і іншими питаннями, пов'язаними з динамікою окремої клітини.

Ця модель є дорогою для реалізації. Для оцінки 0.1 мс часу по цій моделі - потрібно 120 «операцій із плаваючою точкою» (при допущенні, що на кожну експоненту йде тільки 10 операцій). Отже, одержуємо 1200 операцій/1 мс. Отже, можна використати модель Ходжкіна-Хакслі тільки для імітаційного моделювання невеликої кількості нейронів, або коли час моделювання не важливий.

3.4. Нейрокомп'ютерні властивості нейронів

Більшість нейронів є збудливими; тобто вони перебувають у спокої, але при стимуляції можуть розрядитися спайками (нейрон генерує «спайк» = потенціал дії = імпульс напруги). Щоб випробувати цю властивість нейронів – нейрофізіологи вводять імпульси постійного струму через електрод,

прикріплений до нейрона, і реєструють його мембранний потенціал. Вхідний струм й відгук нейрона – звичайно малюється на графіку один під іншим. Поки є вхідний імпульс, нейрон продовжує розряджатися послідовністю спайків. Цей тип поведінку називаються «тонічні спайки»; його можна спостерігати в трьох типах кортикальних нейронів: регулярно-спайкові (regular spiking, RS), збудливі нейрони, низькопорогово-спайкові (low-threshold spiking, LTS), і швидко-спайкові (fast spiking, FS) гальмівні нейрони. Якщо такий нейрон безупинно розряджається - то це означає, що в цей час є вхідний імпульс.

(A) tonic spiking

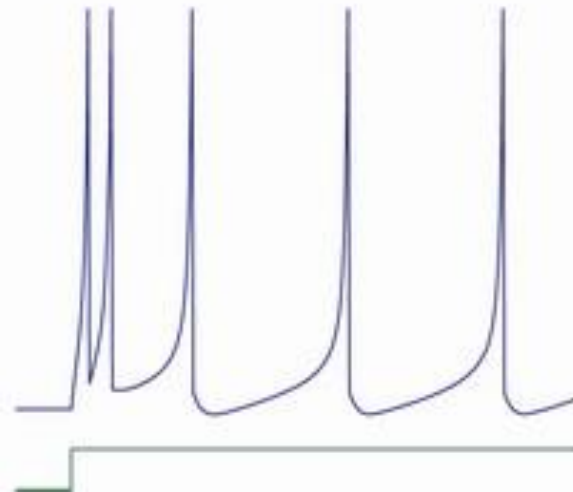


Рисунок 3.2. Тонічні спайки.

Нейрон може розрядитися тільки одним спайком. Такий відгук називається «фазичний спайк», і він корисний для виявлення початку стимуляції.

(B) phasic spiking

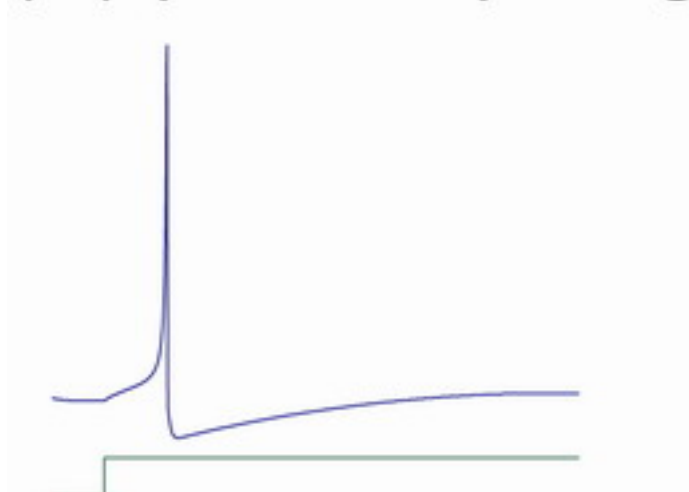


Рисунок 3.3 Фазичні спайки.

Деякі нейрони – такі, як вібруючі нейрони в неокортексі кішки, при їхній стимуляції розряджаються періодичними залпами спайків.

Міжзалпова частота може досягати 50 Гц, і вважається, що такі нейрони роблять внесок у мозкові коливання гамма-частоти.

(C) tonic bursting

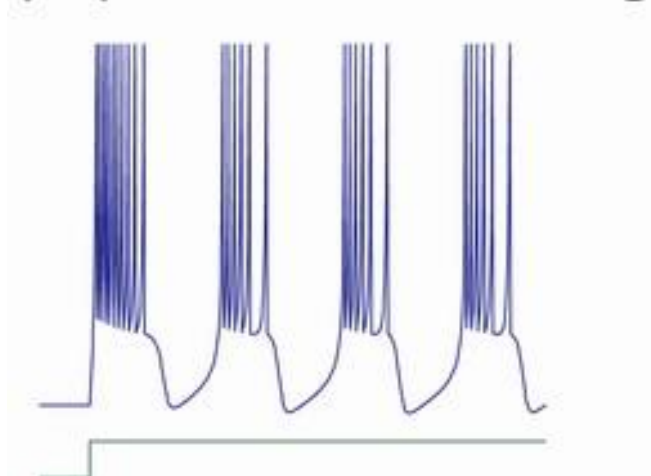


Рисунок 3.4. Тонічні залпи.

Аналогічно «фазичним спайкам», деякі нейрони генерують фазичні залпи. Деякі нейрони повідомляють про початок стимуляції шляхом передачі залпу.

Є три основні гіпотези про важливу роль залпів у мозку, а саме:

- 1) залпи потрібні, щоб перебороти відмову синаптичної передачі, і знизити нейронний шум;
- 2) залпи можуть передавати рельєф вхідного сигналу, тому що вплив залпу на постсинаптичний нейрон сильніше, ніж вплив одного спайка;
- 3) залпи можна використати для виборчого зв'язку між нейронами, причому міжзалпова частота кодує канал зв'язку.

Гарна модель кортикальної нейронної мережі не може не враховувати залпові нейрони.

Внутрішньо-залпові (intrinsically bursting, IB) збудливі нейрони в неокортексі ссавців можуть демонструвати змішаний тип спайкової активності. Вони розряджаються фазичним залпом на початку стимуляції, і потім перемикаються в режим «тонічних спайків». Неясно, який тип обчислень може виконувати такий нейрон, крім виявлення початку стимуляції й повідомлення про тривалість стимуляції.

Найпоширеніший тип «збудливих нейронів» у неокортексі ссавців, а саме регулярно-спайкова (RS) клітина, яка розряджається тонічними спайками зі зменшеною частотою. Тобто, на початку стимуляції частота висока, а потім відбувається адаптація. Низькопорогово-спайкові (LTS) гальмівні нейрони теж мають цю властивість, міжспайкова частота таких клітин може кодувати час, що минув після початку вхідного сигналу.

Деякі нейрони не можуть розряджатися послідовностями спайків низкою частоти. Тобто, вони або перебувають у спокої, або розряджаються послідовністю спайків певної відносно високої частоти, наприклад, 40 Гц. Такі нейрони називаються «нейронами зі збудливістю 2-го класу». Їхня частота розряду є поганим провісником сили стимуляції.

Більшість кортикальних нейронів розряджаються спайками не відразу, а із затримкою, що залежить від сили вхідного сигналу. Для відносно слабкого, але все-таки такого, що перевищує поріг, сигналу - ця затримка (названа також «латентністю спайка»), може бути досить великою. RS-клітини в кортексі

савців можуть мати латентність порядку десятків мілісекунд. Такі затримки забезпечують механізм спайкової синхронізації, що дозволяє закодувати силу вхідного сигналу.

Практично кожна мозкова структура має нейрони, здатні демонструвати коливальні потенціали. Частота таких коливань відіграє важливу роль, і такі нейрони діють як смугові фільтри (що ми обговоримо нижче).

Завдяки явищу резонансу, нейрони, що мають коливальні потенціали, можуть вибірково відповідати на вхідний сигнал, що має частотний спектр, схожий із частотою підпорогових коливань.

Такі нейрони можуть реалізувати частотно-модульовані (frequency-modulated, FM) взаємодії та мультиплексування сигналів. Якщо зімітувати нейрон із двома дублетами (дублет = пари спайків), що мають різні міжспайкові частоти, то цей нейрон відповідає тільки на той дублет, частота якого попадає в резонанс із частотою підпорогових коливань. Такі нейрони називаються «резонаторами».

Нейрони без коливальних потенціалів діють як інтегратори.

Вони віддають перевагу високочастотному вхідному сигналу; чим вище частота вхідного сигналу, тим більше ймовірність того, що нейрон розрядиться. Це може виявитися корисним для виявлення співпадаючих або майже співпадаючих спайків.

Коли нейрон одержує, і потім звільняється від гальмуючого вхідного сигналу він може розрядитися пост-гальмуючим спайком (спайком віддачі). Це явище пов'язане з анодним розмикаючим збудженням у збудливих мембранах. Більшість спайкових нейронів може розрядитися у відповідь на короткий гальмуючий вхідний сигнал, тим самим розмикаючи різницю між збудженням і гальмуванням.

Деякі нейрони, у тому числі таламокортикальні клітини, можуть розряджатися пост-гальмуючими залпами. Вважається, що такі залпи роблять внесок у коливання сну в таламокортикальній системі.

Деякі нейрони - можуть демонструвати два стабільні режими роботи: спокій і «тонічні спайки» (або навіть залпи). Збудливий або гальмуючий імпульс може перемикає нейрон між цими двома режимами, тим самим створюючи цікаву можливість бістабільності й короткочасної пам'яті.

Для перемикає з режиму «тонічних спайків» у режим спокою, вхідний імпульс повинен прийти в підходящій фазі коливань, що говорить про важливість синхронізації спайків при такій обробці інформації.

Після розряду спайком, мембранний потенціал нейрона – може демонструвати тривалу «слідову гіперполяризацію» (after-hyperpolarization, АНР), або тривалий «слідовий потенціал» депольаризації (depolarized after-potential, DAP). Такі DAP можуть з'являтися через дендритний вплив; через високотривалі вхідні струми, активовані під час спайка; або через взаємодію підпорогових потенціалозалежних струмів. У кожному разі, такий нейрон має скорочений рефракторний період і стає суперзбудливим.

Нейрони дуже чутливі до коротких стрибкоподібних вхідних сигналів, але можуть не розрядитися у відповідь на сильний, але повільно зростаючий вхідний сигнал. Під час повільного росту вхідного сигналу: у вхідних струмів є досить часу, щоб інактивувати (а у вихідних струмів є досить часу, щоб активувати), тому нейрон акомодується, стає менш збудливим і не може генерувати спайк.

Незвичайною властивістю багатьох таламокортикальних нейронів є те, що вони перебувають у спокої, поки немає вхідного сигналу, але розряджаються, коли гіперполяризуються шляхом гальмуючого вхідного сигналу або подаваного струму. Це відбувається тому, що струм, що подається активує h-струм й інактивує кальцієвий T-струм, що приводить до тонічних спайок.

У відповідь на тривалу гіперполяризацію таламокортикальний нейрон може розрядитися не спайками, а залпами спайків. Вважається, що такі залпи відбуваються під час «веретеноподібних» коливань у таламокортикальній системі; і що вони відіграють важливу роль у ритмах сну.

3.4. Висновки до третього розділу

Біологічні моделі нейронів намагаються описати імітацію реальних нейронів і їх взаємодії в біологічних нервових системах. Існує кілька основних моделей, які використовуються для вивчення нейронної активності, і вони різняться за ступенем деталізації та способом моделювання біологічних процесів.

Моделі Ходжкіна-Хаксли та LIF (Leaky Integrate and Fire) є найбільш відомими і широко використовуваними для моделювання біологічної активності нейронів.

Спайкові нейронні моделі (SNN) набувають популярності через здатність моделювати тимчасову динаміку та враховувати спайкову активність нейронів.

Моделі нейронів з нейропластичністю дають змогу вивчати механізми навчання та пам'яті в нервових мережах, що робить їх дуже важливими для досліджень.

Вибір моделі залежить від того, наскільки детально потрібно відтворювати біологічні процеси, та від обчислювальних ресурсів, які доступні для моделювання.

4. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ МЕРЕЖІ НЕЙРОНІВ

4.1. Порівняння моделей нейронів спеціальних видів за параметрами

Жодна з моделей не може одночасно демонструвати всі нейрокомп'ютерні можливості. Багато із властивостей є взаємовиключними. Наприклад, нейрон не може бути одночасно інтегратором і резонатором.

Еквівалентна схема нейрона інтегратора може виглядати наступним чином https://www.researchgate.net/figure/Schematic-of-a-the-leaky-integrator-and-b-the-firing-neuron_fig6_3450686.

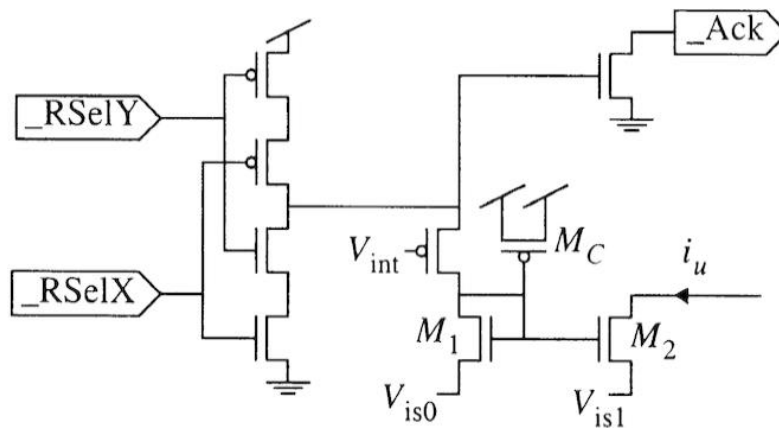


Рисунок 4.1. Нейрон інтегратор.

Тобто фактично існує три представлення нейрона, по перше це біологічний зразок, який має відповідні характеристики. Тут можна досліджувати обчислювальні можливості саме біологічної нейронної мережі, але це досить складна задача на теперешній час, що пов'язано з роботою з живим середовищем, а це дуже нестабільна система. Хоча саме такий підхід має перспективу отримання якісно нових технологій обчислень, якщо використати саме це слово, атже це все ж така інший спосіб розв'язання прикладних задач.

Наступне представлення це апаратна імітація функціонування нейрону, та відповідно нейронної мережі. Приклад такого представлення приведено на рисунку 4.1. Фактично мова іде про побудову DSP-процесору, що працює як нейронна мережа. Саме тут виникають потреби у розробці аналогів обчислювальних елементів, що виконують специфічні операції.

І третє представлення, це математична модель, що реалізована на класичній обчислювальній архітектурі фон Неймана. Тобто така суперечлива концепція, з одного боку намагаємося застосувати технологію, що базується на біологічних принципах, а з іншого отримуємо всі недоліки класичної архітектури.

Отже виникає необхідність розуміти як співвідноситься обчислювальні можливості, що потрібні для реалізації моделі на класичній архітектурі, реалізованих, як різновид архітектури x86, чи навіть DSP, та саме результати моделювання.

Є такі моделі, які можна настроїти на демонстрацію кожної з обчислювальних властивостей.

Далі наводиться порівняльна таблиця, у якій порівнюються моделі нейронів спеціальних видів за параметрами, які було описано у попередньому розділі.

Кількість операцій (FLOPS) у цій таблиці – це оціночна кількість операцій із плаваючою крапкою, що відносяться саме до обчислювальної архітектури фон Неймана та її x86, чи більш точніше x386 реалізації, необхідних, щоб виконати імітаційне моделювання по моделі протягом тимчасового інтервалу 1 мілісекунда.

Слід зауважити, що такого роду оцінка це початкова точка відліку, атже нецрони треба поєднати у мережі та виконати супутні дії, крім того, час виконання моделювання може бути значно більше мілісекунди, наприклад якщо ставиться задача пошуку періодичних частотних процесів, то необхідно проаналізувати декілька періодів.

Порівняння нейрокомп'ютерних властивостей моделей.

МОДЕЛІ:	Кількість FLOPS
integrate-and-fire (інтегратор і розряд)	5
quadratic integrate-and-fire (квадратичний інтегратор і розряд)	7
Izhikevich	13
FitzHugh-Nagumo	72
Hindmarsh-Rose (Хайдмарша - Роуза)	120
Morris-Lecar	600
Hodgkin-Huxley (Ходжкина - Хаксли)	1200

У таблиці преведені оцінки для моделей, таким чином, щоб продемонструвати зріст складності моделі і відповідно пропорційний зріст обчислювальних затрат.

Відповідно тут виникає питання яку модель обрати, чи зосередитись на більш відповідних біологічним системам процесам та витратити ресурси, чи зосередитися безпосередньо на одному чи парі параметрів, які продемонструють суттєві характеристики нейрону та відповідно суттєву поведінку мережі нейронів, що побудована саме з таких одиниць.

При розробці великомасштабних моделей мозку, що складаються із спайкових нейронів, треба знайти компроміс між двома взаємовиключними вимогами: модель одиночного нейрона повинна бути легко обчислюваною і проте вона повинна добре моделювати активність біологічного нейрона.

Використовувати біофізично точні моделі, побудовані по типу моделі Ходжкина-Хакслі, доцільно з обчислювальної точки зору, лише тоді, коли ми можемо з'імітувати роботу деякої обмеженої кількості нейронів в режимі реального часу.

З другого боку, використання моделі інтегратор-та-розряд ефективно з обчислювальної точки зору, проте отримана модель буде занадто простою і вона не зможе генерувати спайкову і залпову активність, яку можна спостерігати в кортикальних нейронах.

Відмовившись від багатьох біофізичних властивостей, властивих моделям Ходжкина-Хакслі, маємо можливість перейти до двовимірного простору, та до системи диференціальних рівнянь наступного виду.

$$\begin{aligned} v' &= xv_2 + yv + z - u + I \\ u' &= a(bv - u) \end{aligned} \quad (*)$$

із додатковим послеспайковим скидом.

$$\text{якщо } v \geq 30 \text{ мВ, то } \left\{ \begin{array}{l} v \leftarrow c \\ u \leftarrow u + d \end{array} \right\}$$

де v і u – змінні; a, b, c, d – коефіцієнти; похідна в цих рівняннях береться за часом t , x, y, z – параметри, що визначаються у обчислювальному експерименті.

Змінна v відповідає мембранному потенціалу нейрона, а u – регенерації мембрани, яка відповідає за активацію калієвого струму і інактивацію натрієвого струму, що забезпечує скид в v . Після того, як спайк досягає піку, v і u змінюють свої значення. Синаптичний струм або поданий постійний струм передається змінній I .

Параметр a описує тимчасову шкалу дії змінної u . Менші значення дають повільніше відновлення. Типове значення $a=0.02$.

Параметр b описує чутливість змінної u в підпороговій нестійкості мембранного потенціалу v . Великі значення комбінації v і u дають кращий результат в можливих підпорогових коливаннях і низькопорогової спайкової динаміці. Типове значення $b=0.2$.

Параметр c описує післяспайковий скид мембранного потенціалу v , обумовлений швидкою надпороговою провідністю K^+ . Типове значення для цього параметру $c = -65$ мВ.

Параметр d описує післяспайковий скид змінної u , обумовлений низькою надпороговою Na^+ і K^+ провідністю. Типове значення для цього параметру $d = 2$ [16].

Така мережа демонструє асинхронну динаміку, подібну кортикальній (ланцюг спайків із частотою близько 8 Гц). Мережа зв'язана випадковим чином і в ній немає синаптичної гнучкості, нейрони самоорганізуються в групи і показують колективну ритмічну поведінку.

Параметри моделі x, y, z у моделі Іжекевича визначаються емпірично, що є суттєвим недоліком обґрунтування.

Отже розуміючи ідею моделі та розуміючи діапазон у яких знаходяться параметри у фізичному світі, можна провести обчислювальний експеримент та отримати оптимальне значення цих параметрів на основі чисельних розрахунків.

Спочатку у середовищі MatLab реалізуємо модель з відомими параметрами, що дозволить протестувати працездатність моделі та її реалізації.

Тестове виконання свідчить про адекватність розробленого коду, та реалізацію, що відповідаю відомим результатам.

Таким чином, можна оцінити обчислювальні можливості елемента для проведення експерименту та зрозуміти фізичну можливість проведення обчислень у випадку мережі. І хоча час виконання сильно залежить від конкретного обчислювального пристрою є декілька ефективних шляхів для прискорення обчислень. Наприклад це розпаралелювання обчислювального процесу чи застосування хмарних рішень, чи об'єднання цих підходів.

4.2. Імітування активності нейронів

Схема обчислень за моделлю (*) виглядає наступним чином.

1. Задаємо початкові параметри
2. Організуємо три вкладених цикли за шуканими параметрами із визначеним у попередньому розділі кроком
3. Внутрі циклу обчислюємо активність мережі на заданому інтервалі часу
4. Внутрі циклу виводимо результат

Тут присутні два принцові моменти.

Перший це організація перебору значень параметрів. Параметри належать діапазону, який сформовано на базі постулату про фізичні властивості середовища і не дуже відповідають ідям організації обчислювального процесу з точки зору оптимальності, але у даному випадку акцент робиться на достовірності. Але при розвитку обчислювального експерименту, на основі отриманих даних, можна буде перейти вже до скороченого інтервалу вірогідно з меншим кроком.

Другий це збереження та аналіз результатів моделювання, це масив, чи його графічне представлення, що потребує великого простору для зберігання, підтримки структури зберігання та організації методів аналізу як ручними засобами так і машинними.

Далі приведено фрагмент ключового коду, що здійснює моделювання активності нейронної мережі за системою (*) з фіксованими шуканими параметрами. Код призначений для виконання у середовищі МатЛаб.

Містить дві ключові частини, а саме роботу з моделлю (*), як моделлю нейрону, а точніше електричної активності та побудову мережі нейронів з цих елементів та відслідковування активності всієї мережі, як одного цілого.

```

L=800;
N=200;
re=rand(Ne,1);
ri=rand(Ni,1);
a=[0.02*ones(Ne,1); 0.02+0.08*ri];
b=[0.2*ones(Ne,1); 0.25-0.05*ri];
c=[-65+15*re.^2; -65*ones(Ni,1)];
d=[8-6*re.^2; 2*ones(Ni,1)];
S=[0.5*rand(Ne+Ni,Ne), -rand(Ne+Ni,Ni)];
v=-65*ones(Ne+Ni,1);
u=b.*v;
firings=[];
for t=1:1000
l=[5*randn(Ne,1);2*randn(Ni,1)];
fired=find(v>=30);
firings=[firings; t+0*fired,fired];
v(fired)=c(fired);
u(fired)=u(fired)+d(fired);
l=l+sum(S(:,fired),2);
v=v+0.5*(0.04*v.^2+5*v+140-u+l);
v=v+0.5*(0.04*v.^2+5*v+140-u+l);
u=u+a.*(b.*v-u);
end;
plot(firings(:,1),firings(:,2),'.');

```

Програмний продукт що, імітує активність спайкових та залпових нейронів написано при використанні програмного середовища MatLab 14.0, тому перед початком роботи з програмним продуктом треба запустити MatLab.

Після запуску MatLab 14.0 треба відкрити файл l.m, скориставшись меню «File» далі «Open». Після цього автоматично виконається підпрограма «MatLab Editor», у головному вікні якій з'явиться код програми.

Далі треба натиснути «F5» або скористатись меню «Debug» далі «Run». Коли буде виконано ці дії, з'явиться головна форма програми.

Для припинення роботи системи як додатку необхідно натиснути кнопку «Quit».

Для того, щоб призупинити роботу треба натиснути кнопку «Pause».

В лівому нижньому куті розташовано модуль вибору нейрокомп'ютерних властивостей. Цей модуль вміщує 10 варіантів моделей.

Кожна кнопка позначена літерою латинської абетки. Ці літери відповідають властивостям, які описано у четвертому розділі магістерської роботи.

Обравши необхідну нейрокомп'ютерингову властивість (натиснувши відповідну кнопку) користувач має змогу побачити (й зкоригувати при необхідності) параметри, які впливають на нейроактивність. Ці параметри відображаються у модулі «Parameters». Також цей модуль містить кнопку «current», натиснувши яку користувач активує процедуру імітації безперервної подачі постійного електричного струму.

Натиснувши на кнопку «Короткий імпульс» користувач імітує короткочасний позитивно спрямований імпульс току, який подається на вход до нейрону.

Натиснувши на кнопку «i» користувач імітує короткочасний негативно спрямований імпульс току, який подається на вход до нейрону.

Натиснувши на кнопку «Початок» і клікнув на полі «Фаза» користувач ініціює зміну початкових параметрів системи, а саме мембранного потенціалу та змінної розряду.

На полі «phaset» відображаються значення мембранного потенціалу та змінної розряду.

Повзунок «Швидкість процесу» змінює швидкість імітування, тут мають значення параметри системи на якій виконується моделювання.

Повзунок «Сила струму» змінює силу струму, який подається на вхід нейрона.

На графіку 4.1 відтворюється зміна струму, який подається на вхід нейрона з пливом часу.

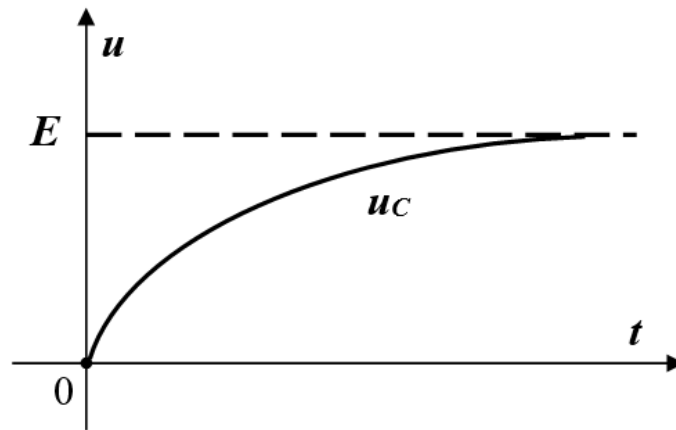


Рисунок 4.2. - Графік струму

На графіку 4.2. в режимі реального часу відтворюється симуляція активності нейрона (накоплення і скид заряду на мембрані нейрона з пливом часу).

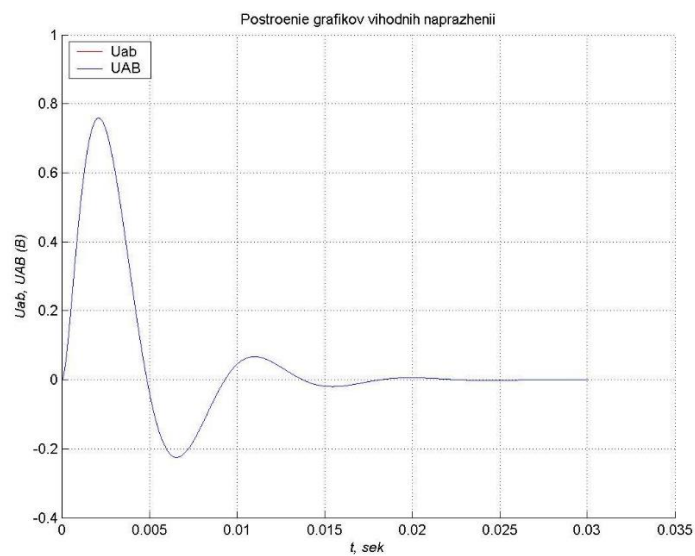


Рисунок 4.3. - Графік активності нейрона.

Порівнюючи результати з відомими отримаємо підтвердження коректності реалізованого алгоритму.

Далі визначаємо простір пошуку параметрів моделі, відповідно до фізичних властивостей середовища визначаємо інтервали

$$z \in [0, \dots, 0,01]$$

$$y \in [0, \dots, 10]$$

$$z \in [0, \dots, 1000]$$

відповідно до обчислювальних можливостей обираємо шаг для моделювання

для x шаг 0.001,

для y шаг 1,

для z шаг 1

треба зазначити, що шаг можна зменшити і отримати більш точні значення параметрів, другий момент самі діапазони, ситуація коли параметри близькі від кінців відрізка вочевидь не буде відповідати фізиці процесів, але повинна бути визначеність, на обчислювальний процес вплив буде не значний.

Таким чином потрібно провести 100000 моделювань та обрати оптимальний результат.

Як зазначено у таблиці 4.1, модель вимагає 13 операцій для реалізації ітерації, отже складність обчислювальної процедури можна оцінити 1,3 мільйони базових операцій.

Далі виникає питання критерію оптимальності для обрання найкращих параметрів.

Тут можна піти двома шляхами, це безпосередньо аналіз отриманих графіків, чи відповідність чисельному критерію.

Для даного випадку більш підходить безпосередній аналіз, але проаналізувати сто тисяч графіків досить складно, отже вимушено будемо орієнтуватися на відповідність емпіричному параметру. Хоча певну кількість графіків у певній області проаналізувати для уточнення можливо. Відповідно потрібно мати деяку систему відображення багатомірного представлення, на базі якої можна проводити відповідний аналіз.

Далі можна виконувати повноцінний обчислювальний експеримент з заданими цілями та очікуванням отримати додаткові знання про мережу та процес обчислень.

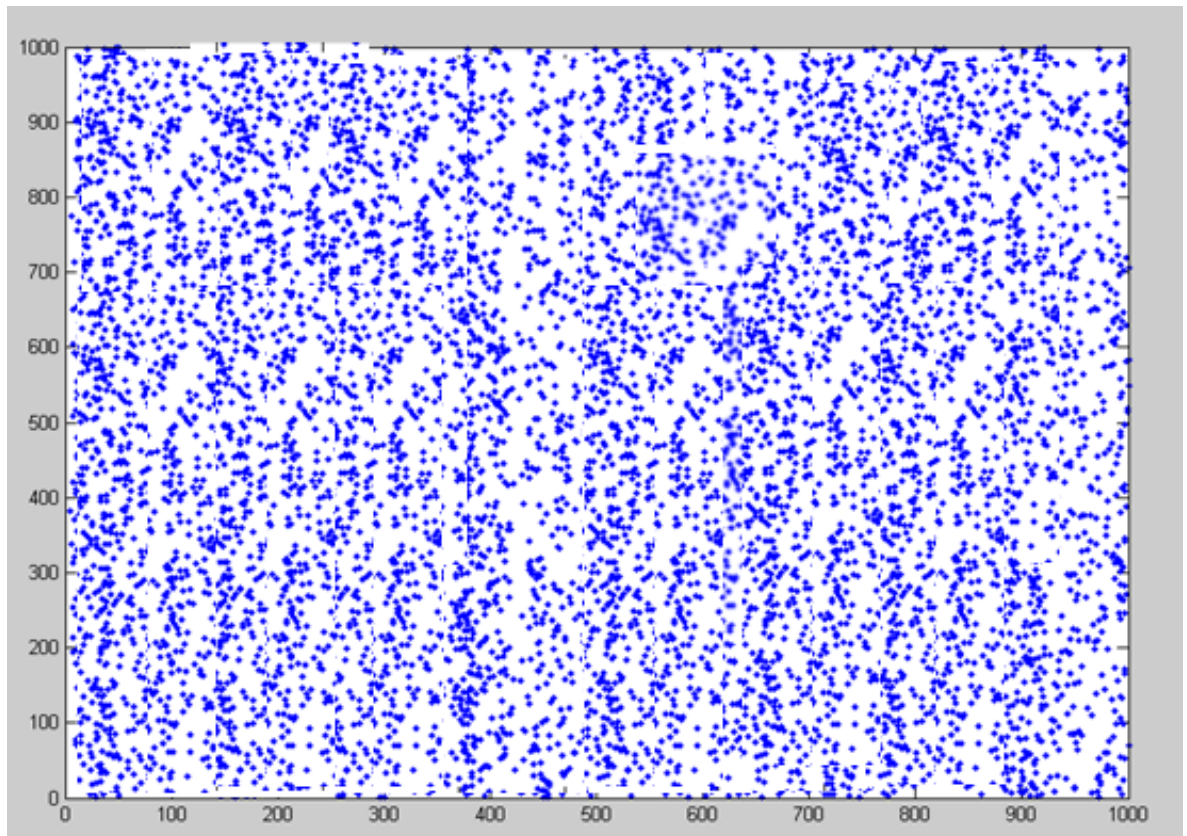


Рисунок 4.4. - Проміжний результат моделювання

На рисунку 4.4 приведено приклад випадку, коли активність мережі проявляє себе випадковим чином, тобто виникають сполохи активності у мережі, з одного боку вона якось функціонує, а з іншого виділити якусь структуру у поведінці неможливо.

Також можна побачити ситуацію, коли активність мережі стимулювати не вдається, що пов'язано з малими токами активації, що відповідають невеликим значенням змінних параметрів.

Однією з проблем аналізу системи з шуканими параметрами, як раз і є велика кількість побудованих моделей, які апріорно не матимуть проявлення активності мережі, чи будуть перезбуджувати мережу та фактично її перевантажувати, зупиняючи передачу імпульсів у змінному вигляді.

Але відмовитися від надлишкових обчислень у рамках запропонованої роботи саме з моделлю немає можливості, тому, що обґрунтування діапазону змін параметру моделі базується на фізичній інтерпретації.

Мова тут іде про те, що для виникнення активності потрібна хоч якась стимуляція, відповідно повинна виникнути різниця потенціалів, яка на даному етапі визначитись можна тільки оціночно, отже потрібно використовуватити нижню оцінку, а вона визначається початком стимуляції, тобто початкове значення це нуль. І така ситуація коректна для всіх трьох параметрів.

Чи можна проводити обчислювальний процес за іншою схемою, наприклад відштовхуюсь від експертно визначених параметрів та працюючи у деякому супутньому просторі, напевно це можливо, але це вимагатиме обґрунтувань, яких апріорії немає, хоча можуть бути отримані апостеріорно. Тобто при обмежених знаннях про систему можна пошук обґрунтувань відкласти, а замість цього провести дослідження значень параметрів у вьому припустимому діапазоні та отримати обґрунтування, що базуються на результатах обчислювального експерименту, що є цілком придатним науковим результатом.

Аналіз отриманих зображень дозволив виділити моделі, які більш за все відповідають шуканим критеріям.

Неявний отриманий результат, це монотоність функцій змінних параметрів у часі. Неявний тому, що немає формальної можливості обґрунтування, але емпірично у цьому немає сумнівів. Тут виникає нечітка ситуація, коли можна зрозуміти як модифікувати обчислювальний процес, але немає формального обґрунтування таких дій. Що теж можна вважати цікавим результатом експерименту.

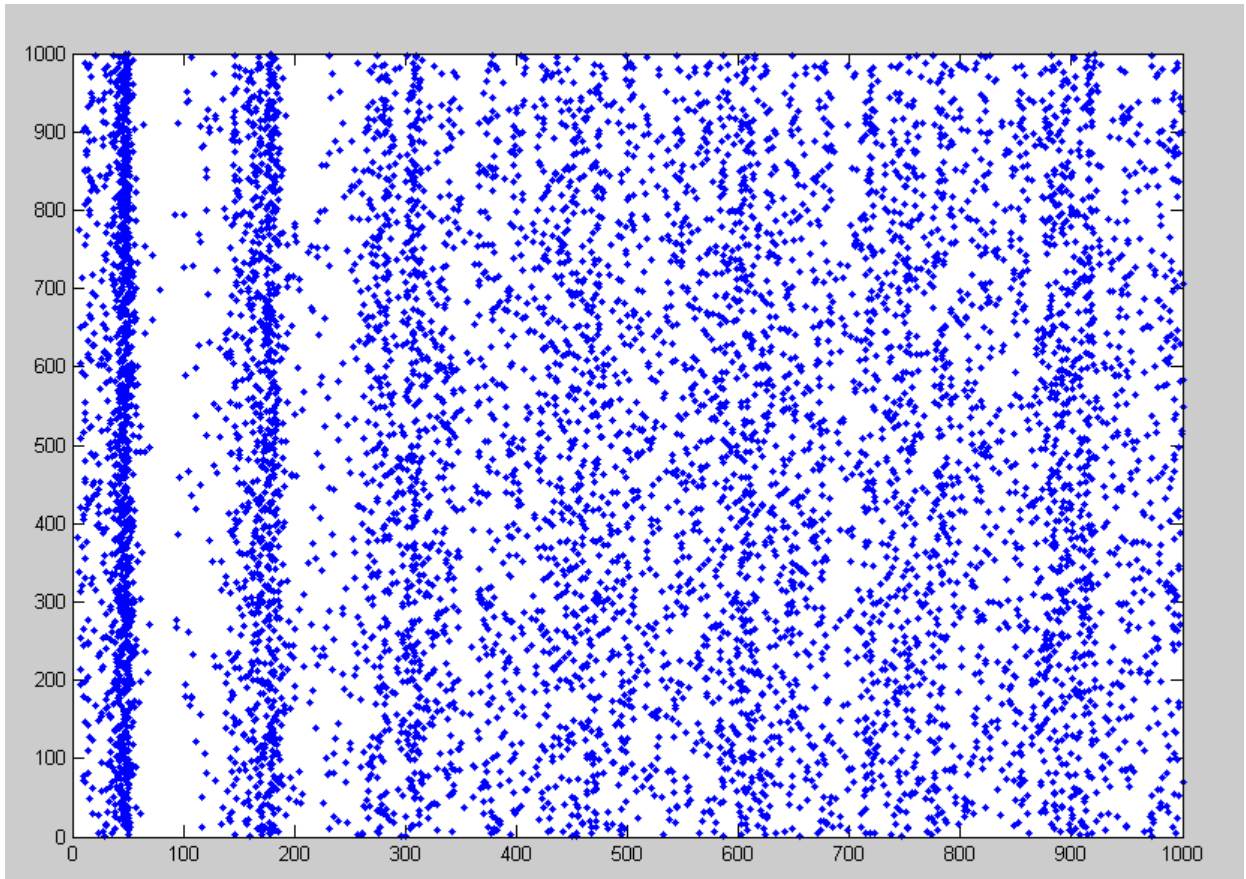


Рисунок 4.5. - Кінцевий результат моделювання

На отриманому графіку бачимо розповсюдження нервових імпульсів вздовж нервової мережі. Розповсюдження йде хвилями, які таять із часом. Стимуляція початкового слою мережі нейронів виконується “правильним” током у випадкові моменти часу для кожного з нейронів. Ключовим моментом для оцінки функціонування моделі є частота хвиль, саме вона є критерієм вибору графіка та відповідно параметрів, які відповідають саме такій картині.

Для системи, результат функціонування якої зображена на рисунку, параметри моделі мають значення $x=0,004$, $y= 5$, $z=137$. Це і є основний результат моделювання.

4.3. Висновки до четвертого розділу

В четвертому розділі кваліфікаційної роботи проведено імітаційне моделювання мережі нейронів за допомогою розробленого програмного забезпечення на основі запропонованих в кваліфікаційній роботі обчислювальних методів.

Деяка різниця між параметрами отриманої моделі та параметрами запропонованими Іжекевичем, який вважає їх адекватними, може бути пояснена налаштуваннями критеріїв відповідності, але може свідчити і про більш тонке налаштування. Це відповідно означає необхідність проведення більш масштабного експерименту з меншими шагами параметрів та уточненням параметрів оптимуму.

Таким чином проаналізовано моделі спайкових нейронів, як поведінка нейронів залежить від вимірюваних фізіологічних параметрів (таких, як максимальні провідності, (ін)активаційні функції й тимчасові константи сталих режимів, за допомогою моделі Ходжкіна – Хакслі, але у реальному часі можна зімітувати тільки десятки зв'язаних спайкових нейронів (у залежності від обчислювальних можливостей).

Зімітувати сотні «спайкових нейронів» у реальному часі з роздільною здатністю 1 мс, можливо з «моделлю I&F». Однак, ця модель не може демонструвати навіть найбільш фундаментальних властивостей кортикальних спайкових нейронів, і із цієї причини цю модель потрібно завжди по можливості уникати. Єдиною перевагою «моделі I&F» є те, що вона лінійна, і тому добре піддається математичному аналізу. Квадратична I&F модель практично так само ефективна, як і лінійна, і до того ж демонструє багато важливих властивостей дійсних нейронів (такі, як спайки з латентністю, бістабільністю режимів спокою й «тонічних спайків»). Однак, ця модель є одномірною (1-D), отже, вона не може демонструвати залпи й адаптацію частоти.

Тому, цю модель можна використати в імітаційному моделюванні кортикальних нейронних мереж тільки тоді, коли біологічна доречність не є важливим критерієм.

Ця модель була використана для імітаційного моделювання повнозв'язної мережі з 1000 кортикальних спайкових нейронів, у реальному часі з роздільною здатністю 1 мс, - використовуючи тільки наявні обчислювальні ресурси. Ця мережа демонструвала ритми в альфа- і гама - частотному діапазоні, перехідну й сталу синхронність спайків.

Кожен нейрон бере участь у безлічі полісинхронних груп; тому загальне число груп, які здатна запам'ятати ця модель може значно перевершувати кількість нейронів, або навіть кількість синапсів мережі. Це означає наявність вторинного результату моделювання. Та можливість проведення досліджень саме у напрямці використання мережі, як оперативної пам'яті.

Тобто обчислювальний експеримент дозволив продемонструвати активність мережі нейронів, як цільної системи, що побудована з елементів – моделей нейронів, що побудовані на базі системи (*), продемонстрував принципову можливість обчислювальним, а не емпіричним шляхом отримання параметрів моделі нейрона. Що у свою чергу підтверджує можливість ефективного використання обчислювальних схем для доведення теоретичних положень

ВИСНОВКИ

У роботі проаналізовано сучасні обчислювальні технології і зроблено висновок про обмеження класичної архітектури фон Неймана. Не дивлячись на технічні рішення, які дозволяють розширити обмеження, все рівно є певний поріг обчислювальних можливостей і з історичного досвід він буде зміщуватись але не зникати.

Далі проаналізовано наступна обчислювальна технологія – нейроні мережі. У них є особливість, а саме те що маючи великі можливості у логіці обробки даних, алгоритм реалізується на архітектурі фон Неймана, тобто на неприродній для мережі архітектурі. Крім того сама архітектура фон Неймана має різновиди реалізації, що ще ускладнює можливі реалізації. Але роботи у цьому напрямку виконуються саме тому, що існуючі результати розв'язання прикладних задач за допомогою нейронних мереж дуже високі навіть на не ефективній обчислювальній схемі.

Інший підхід при побудові обчислювальних нейронних мереж базується на побудові справжньої мережі з нейронів. Тут вже є варіанти, чи використання спеціалізованих процесорів у якості нейронів, чи нейропроцесорів, чи використання живих біологічних структур. Тут виникає потреба у розумінні того як працюють біологічні нейрони та які параметри цього функціонування можуть бути використані у обчислювальному процесі і за рахунок відповідно яких ресурсів.

У роботі розглянуто базові моделі нейронів та реалізації їх моделей у МатЛаб.

Приведені оцінки складності виконання деяких з цих моделей на обчислювальній техніці та зроблено висновки про сфери використання і баланс між результатами моделювання та затратами на обчислення.

Реалізовано модель нейрону, що задається системою диференціальних рівнянь (*).

Порівнюючи результати з відомими отримано підтвердження коректності реалізованого алгоритму моделювання.

Параметри моделі x, y, z у моделі (*) визначаються емпірично, що є суттєвим недоліком обґрунтування.

Визначено простір пошуку параметрів моделі, відповідно до фізичних властивостей середовища.

Отже розуміючи ідею моделі та розуміючи діапазон у яких знаходяться параметри у фізичному світі, було проведено обчислювальний експеримент та отримано умовно оптимальне значення цих параметрів на основі чисельних розрахунків.

Зроблено висновки про принципи моделювання, які дають змогу за рахунок великих обчислювальних ресурсів аналізувати багато варіантів реалізації систем, що дозволяє потім отримати оптимальні параметри моделей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Винер Н., Розенблют А.. Проведение импульсов в сердечной мышце. Математическая формулировка проблемы проведения импульсов в сети связанных возбудимых элементов, в частности в сердечной мышце. Кибернетический сборник 3, ИЛ, 1961.
2. Гельфанд И. М., Цетлин М. Л.. О континуальных моделях управляющих систем. ДАН СССР 131, 6 (1960), стр. 1242—1245.
3. Модели структурно-функциональной организации некоторых биологических систем. Сб. статей, 'Наука', 1966.
4. Беркинблит М. Б., Дунин-Барковский П. Л.. Распространение последовательности импульсов в одномерной возбудимой среде. Биофизика 14, 2 (1969), стр. 322—326.
5. Беркинблит М. Б., Фомин С. В., Холопов А. В.. Распространение импульсов в одномерной возбудимой среде. Биофизика 11, 2 (1966), стр. 329—336.
6. Вычислительные устройства в биологии и медицине. Сб. статей, 'Мир'. 1967.
7. Чайлахян Л. М.. Электрическая структура возбудимых тканей и механизм проведения нервного импульса. Биофизика, 7, 5 (1962), стр. 639—651.
8. Катц Б.. Нерв, мышца и синапс. 'Мир', 1968.
9. Либерман Е. А.. 'Генераторы' и 'насосы' живой клетки. 'Знание', 1965.
10. Ходжкин А.. Нервный импульс. 'Мир', 1965.
11. Ходоров Б. И.. Проблема возбудимости. 'Медицина', 1969.
12. Беркинблит М. Б., Ковалев С. А., Смолянинов В. В., Чайлахян Л. М.. Электрическое поведение миокарда как системы и характеристики мембран клеток сердца. в сб. [3], 1966, стр. 71—111.
13. Маркин В. С., Чизмаджев Ю. А.. О распространении возбуждения в одной модели нервного волокна. Биофизика 12, 5 (1967), стр. 900—907.

14. Ходоров Б. Н., Тимин Е. П.. II. Теоретический анализ механизмов проведения нервного импульса по неоднородному волокну. III. Трансформация ритмов в охлажденном участке волокна. Биофизика 15, 3 (1970), стр. 503—512.
15. Eugene M. Izhikevich, IEEE Transactions on neural networks, vol. 14, NO. 6, NOVEMBER 2003
16. Eugene M. Izhikevich, IEEE Transactions on neural networks, vol. 15, NO. 5, San-Diego, 2004

ДОДАТОК А

Тонічні спайки

```

%%%%%%%%%% (A) tonic spiking %%%%%%%%%%%
subplot(5,4,1)
a=0.02; b=0.2; c=-65; d=6;
V=-70; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:100;
T1=tspan(end)/10;
for t=tspan
    if (t>T1)
        I=14;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 max(tspan)],-90+[0 0 10 10]);
axis([0 max(tspan) -90 30])

```

```
axis off;
title('(A) tonic spiking');
```

Фазичні спайки

```
%%%%%%%%%%%%% (B) phasic spiking %%%%%%%%%%%%%%
subplot(5,4,2)%
a=0.02; b=0.25; c=-65; d=6;
V=-64; u=b*V;
VV=[]; uu=[];
tau = 0.25;tspan = 0:tau:200;
T1=20;
for t=tspan
    if (t>T1)
        I=0.5;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 max(tspan)],-90+[0 0 10 10]);
axis([0 max(tspan) -90 30])
```

```
axis off;
title('(B) phasic spiking');
```

Тонічні залпи

```
%%%%%%%%%%%%% (C) tonic bursting %%%%%%%%%%%%%%
subplot(5,4,3)
a=0.02; b=0.2; c=-50; d=2;
V=-70; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:220;
T1=22;
for t=tspan
    if (t>T1)
        I=15;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 max(tspan)],-90+[0 0 10 10]);
```

```
axis([0 max(tspan) -90 30])
```

```
axis off;
```

```
title('(C) tonic bursting');
```

Фазичні залпи

```
%%%%%%%%%%%%% (D) phasic bursting %%%%%%%%%%%%%%
```

```
subplot(5,4,4)
```

```
a=0.02; b=0.25; c=-55; d=0.05;
```

```
V=-64; u=b*V;
```

```
VV=[]; uu=[];
```

```
tau = 0.2; tspan = 0:tau:200;
```

```
T1=20;
```

```
for t=tspan
```

```
    if (t>T1)
```

```
        I=0.6;
```

```
    else
```

```
        I=0;
```

```
    end;
```

```
    V = V + tau*(0.04*V^2+5*V+140-u+I);
```

```
    u = u + tau*a*(b*V-u);
```

```
    if V > 30
```

```
        VV(end+1)=30;
```

```
        V = c;
```

```
        u = u + d;
```

```
    else
```

```
        VV(end+1)=V;
```

```
    end;
```

```
    uu(end+1)=u;
```

```
end;
```

```
plot(tspan,VV,[0 T1 T1 max(tspan)],-90+[0 0 10 10]);
```

```
axis([0 max(tspan) -90 30])
axis off;
title('(D) phasic bursting');
```

Змішана модель (залпи, потім спайки)

```
%%%%%%%%%% (E) mixed mode %%%%%%%%%%%
subplot(5,4,5)
a=0.02; b=0.2; c=-55; d=4;
V=-70; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:160;
T1=tspan(end)/10;
for t=tspan
    if (t>T1)
        I=10;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;
end;
```



```

plot(tspan,VV,[0 T1 T1 max(tspan)],-90+[0 0 10 10]);
axis([0 max(tspan) -90 30])
axis off;
title('(E) mixed mode');

```

Адаптація частоти спайків

```

%%%%%%%%%% (F) spike freq. adapt %%%%%%%%%%%
subplot(5,4,6)
a=0.01; b=0.2; c=-65; d=8;
V=-70; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:85;
T1=tspan(end)/10;
for t=tspan
    if (t>T1)
        I=30;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;

```

```

end;
plot(tspan,VV,[0 T1 T1 max(tspan)],-90+[0 0 10 10]);
axis([0 max(tspan) -90 30])
axis off;
    title('(F) spike freq. adapt');
    Збудливість 1-го Класу

%%%%%%%%%%%%% (G) Class 1 exc. %%%%%%%%%%%%%%
subplot(5,4,7)
a=0.02; b=-0.1; c=-55; d=6;
V=-60; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:300;
T1=30;
for t=tspan
    if (t>T1)
        I=(0.075*(t-T1));
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+4.1*V+108-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;

```

```

end;
plot(tspan,VV,[0 T1 max(tspan) max(tspan)],-90+[0 0 20 0]);
axis([0 max(tspan) -90 30])
axis off;
title('(G) Class 1 excitable');

```

Збудливість 2-го класу

```

%%%%%%%%%%%%%% (H) Class 2 exc. %%%%%%%%%%%%%%%
subplot(5,4,8)
a=0.2; b=0.26; c=-65; d=0;
V=-64; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:300;
T1=30;
for t=tspan
    if (t>T1)
        I=-0.5+(0.015*(t-T1));
    else
        I=-0.5;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;

```

```

end;
plot(tspan,VV,[0 T1 max(tspan) max(tspan)],-90+[0 0 20 0]);
axis([0 max(tspan) -90 30])
axis off;
    title('(H) Class 2 excitable');

```

Латентність спайка

```

%%%%%%%%%% (I) spike latency %%%%%%%%%%%
subplot(5,4,9)
a=0.02; b=0.2; c=-65; d=6;
V=-70; u=b*V;
VV=[]; uu=[];
tau = 0.2; tspan = 0:tau:100;
T1=tspan(end)/10;
for t=tspan
    if t>T1 & t < T1+3
        I=7.04;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else

```

```

    VV(end+1)=V;
end;
uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 T1+3 T1+3 max(tspan)],-90+[0 0 10 10 0 0]);
axis([0 max(tspan) -90 30])
axis off;
title('(I) spike latency');

```

Підпорогові коливання

```

%%%%%%%%%%%%% (J) subthresh. osc. %%%%%%%%%%%%%%
subplot(5,4,10)
a=0.05; b=0.26; c=-60; d=0;
V=-62; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:200;
T1=tspan(end)/10;
for t=tspan
    if (t>T1) & (t < T1+5)
        I=2;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else

```

```

    VV(end+1)=V;
end;
uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 (T1+5) (T1+5) max(tspan)],-90+[0 0 10 10 0 0],...
    tspan(220:end),-10+20*(VV(220:end)-mean(VV)));
axis([0 max(tspan) -90 30])
axis off;
title('(J) subthreshold osc.');
```

Перевага по частоті та резонанс

```

%%%%%%%%%%%%% (K) resonator %%%%%%%%%%%%%%
subplot(5,4,11)
a=0.1; b=0.26; c=-60; d=-1;
V=-62; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:400;
T1=tspan(end)/10;
T2=T1+20;
T3 = 0.7*tspan(end);
T4 = T3+40;
for t=tspan
    if ((t>T1) & (t < T1+4)) | ((t>T2) & (t < T2+4)) | ((t>T3) & (t < T3+4)) | ((t>T4)
& (t < T4+4))
        I=0.65;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
```

```

u = u + tau*a*(b*V-u);
if V > 30
    VV(end+1)=30;
    V = c;
    u = u + d;
else
    VV(end+1)=V;
end;
uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 (T1+8) (T1+8) T2 T2 (T2+8) (T2+8) T3 T3 (T3+8) (T3+8)
T4 T4 (T4+8) (T4+8) max(tspan)],-90+[0 0 10 10 0 0 10 10 0 0 10 10 0 0 10 10 0
0]);
axis([0 max(tspan) -90 30])
axis off;
title('(K) resonator');

```

Інтегрування й виявлення збігів

```

%%%%%%%%%%%%% (L) integrator %%%%%%%%%%%%%%
subplot(5,4,12)
a=0.02; b=-0.1; c=-55; d=6;
V=-60; u=b*V;
VV=[]; uu=[];
tau = 0.25; tspan = 0:tau:100;
T1=tspan(end)/11;
T2=T1+5;
T3 = 0.7*tspan(end);
T4 = T3+10;
for t=tspan

```

```

if ((t>T1) & (t < T1+2)) | ((t>T2) & (t < T2+2)) | ((t>T3) & (t < T3+2)) | ((t>T4)
& (t < T4+2))
    I=9;
else
    I=0;
end;
V = V + tau*(0.04*V^2+4.1*V+108-u+I);
u = u + tau*a*(b*V-u);
if V > 30
    VV(end+1)=30;
    V = c;
    u = u + d;
else
    VV(end+1)=V;
end;
uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 (T1+2) (T1+2) T2 T2 (T2+2) (T2+2) T3 T3 (T3+2) (T3+2)
T4 T4 (T4+2) (T4+2) max(tspan)],-90+[0 0 10 10 0 0 10 10 0 0 10 10 0 0 10 10 0
0]);
axis([0 max(tspan) -90 30])
axis off;
title('(L) integrator');

```

Спайк віддачі

```

%%%%%%%%%%%% (M) rebound spike %%%%%%%%%%
subplot(5,4,13)
a=0.03; b=0.25; c=-60; d=4;
V=-64; u=b*V;

```



```

VV=[]; uu=[];
tau = 0.2; tspan = 0:tau:200;
T1=20;
for t=tspan
    if (t>T1) & (t < T1+5)
        I=-15;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 (T1+5) (T1+5) max(tspan)],-85+[0 0 -5 -5 0 0]);
axis([0 max(tspan) -90 30])
axis off;
title('(M) rebound spike');

```

Залп віддачі

```

%%%%%%%%%%%% (N) rebound burst %%%%%%%%%%%%%%
subplot(5,4,14)
a=0.03; b=0.25; c=-52; d=0;
V=-64; u=b*V;

```

```

VV=[]; uu=[];
tau = 0.2; tspan = 0:tau:200;
T1=20;
for t=tspan
    if (t>T1) & (t < T1+5)
        I=-15;
    else
        I=0;
    end;
    V = V + tau*(0.04*V^2+5*V+140-u+I);
    u = u + tau*a*(b*V-u);
    if V > 30
        VV(end+1)=30;
        V = c;
        u = u + d;
    else
        VV(end+1)=V;
    end;
    uu(end+1)=u;
end;
plot(tspan,VV,[0 T1 T1 (T1+5) (T1+5) max(tspan)],-85+[0 0 -5 -5 0 0]);
axis([0 max(tspan) -90 30])
axis off;
title('(N) rebound burst');

```