

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Комп'ютерних наук

## **Пояснювальна записка**

до бакалаврської роботи  
на ступінь вищої освіти бакалавр  
на тему: «**ДОСЛІДЖЕННЯ МЕТОДІВ ШІ ДЛЯ АНАЛІЗУ СТАТИСТИЧНИХ  
ДАНИХ РОЗВИТКУ ВІРУСНОЇ ЕПІДЕМІЇ**».

Виконав: студент 4 курсу, групи КНД–41  
спеціальності 122 Компютерні науки

(шифр і назва спеціальності)

Корнійчик І.Ю.

(прізвище та ініціали)

Керівник Звенигородський О.С

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Київ – 2021

# РОЗДІЛ 1 ОСОБЛИВОСТІ ОРГАНІЗАЦІЇ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРИКЛАДНИХ ЗАВДАНЬ

## 1.1 Що таке нейронні мережі?

Нейронна мережа - це взаємопов'язана збірка простих обробних елементів, блоків або вузлів, функціональність яких вільно базується на нейроні тварини. Обробна здатність мережі зберігається в міцності або взаємоз'єднаних вагах, отриманих в процесі адаптації до набору навчальних моделей або вивчення з них. Щоб трохи розібратися в цьому, спочатку ми швидко переглянемо деякі основні моменти нейробіології. Мозок людини складається із приблизно 100 мільярдів нервових клітин або нейронів, високостилізований приклад яких наведено на Рисунку 1.1. Нейрони спілкуються за допомогою електричних сигналів, які є короткочасними імпульсами або «стрибками» напруги клітинної стінки або мембрани. Міжнейронні з'єднання опосередковуються електрохімічними стику, що називаються синапсами, які розташовані на гілках клітини, які називаються дендритами.

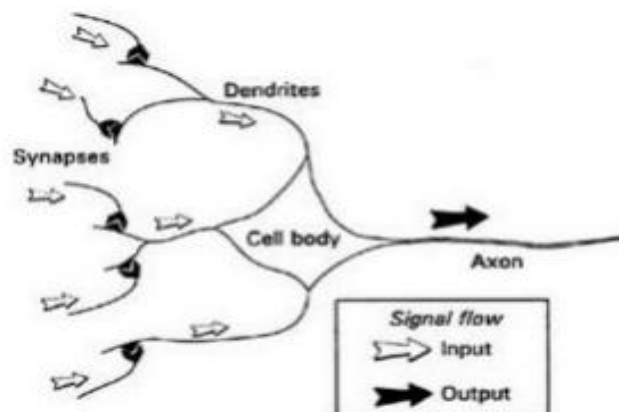


Рисунок 1.1- Основні компоненти нейрона, показані у стилізованій формі.

Кожен нейрон зазвичай отримує багато тисяч з'єднань від інших нейронів і, отже, постійно отримує безліч вхідних сигналів, які з часом потрапляють до клітинного тіла. Тут вони певним чином інтегруються або підсумовуються, і, грубо кажучи, якщо результуючий сигнал перевищує деякий поріг, тоді нейрон буде «спрацьовувати» або генерувати імпульс напруги у відповідь. Потім він передається іншим нейронам через розгалужене волокно, відоме як аксон. Визначаючи, чи слід виробляти імпульс чи ні, деякі вхідні сигнали виробляють гальмівний ефект і, як правило, перешкоджають спрацьовуванню, тоді як інші викликають збудження і сприяють генерації імпульсу. Потім передбачається, що

відмінна обробна здатність кожного нейрона залежить від типу - збудливого чи гальмівного - та сили його синаптичних зв'язків з іншими нейронами. Саме цю архітектуру та стиль обробки ми сподіваємось включити в нейронні мережі, і, зважаючи на наголос на важливості міжнейронних зв'язків, цей тип системи іноді називають коннекціоністським, а вивчення цього загального підходу - коннекціонізмом. Ця термінологія часто зустрічається для нейронних мереж у контексті психологічно натхненних моделей когнітивних функцій людини. Однак ми будемо використовувати його досить загально для посилання на нейронні мережі без посилання на якусь конкретну область застосування.

Штучні еквіваленти біологічних нейронів - це вузли або одиниці у нашому попередньому визначенні, а прототиповий приклад наведено на Рисунку 1.2. Синапси моделюються одним числом або вагою, так що кожен вхід помножується на вагу, перш ніж надсилатися до еквівалента тіла клітини. Тут зважені сигнали підсумовуються за допомогою простого арифметичного додавання о подачі активації вузла. У типі вузла, показаному на Рисунку 1.2, так званий пороговий логічний блок (TLU) - активація потім порівнюється з пороговим значенням; якщо активація перевищує порогове значення, блок видає високоцінний результат (умовно "1"), інакше він видає нуль.

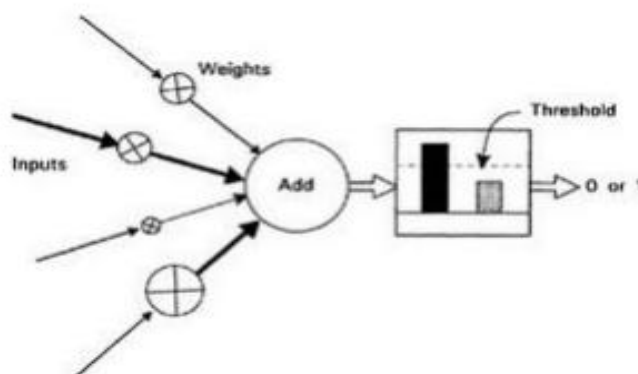


Рисунок 1.2 - Простий штучний нейрон.

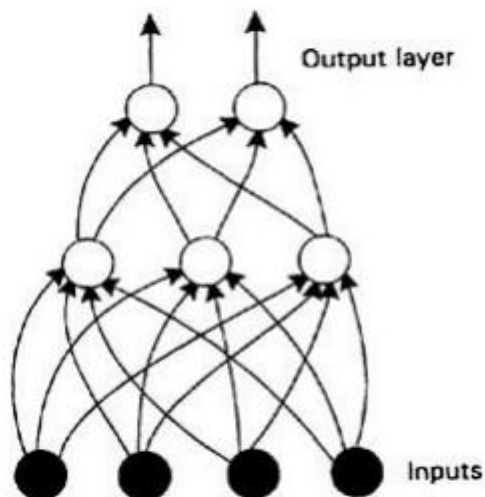


Рисунок 1.3 - Простий приклад нейронної мережі.

На рисунку розмір сигналів представлений шириною відповідних стрілок, ваги зображуються символами множення в колах, а їх значення повинні пропорційні розміру символу; використовувались лише позитивні ваги. TLU - це найпростіша (і історично найдавніша (McCulloch & Pitts 1943)) модель штучного нейрона.

Термін "мережа" буде використовуватися для позначення будь-якої системи штучних нейронів. Це може варіюватися від чогось такого простого, як окремий вузол, до великої колекції вузлів, у яких кожен з них підключений до кожного іншого вузла в мережі. Один тип мережі показаний на Рисунку 1.3. Кожен вузол тепер відображається лише колом, але ваги неявні для всіх з'єднань. Вузли розташовані в багатошаровій структурі, в якій кожен сигнал походить від входу і проходить через два вузли до досягнення виходу, за яким він більше не трансформується. Ця структура переадресації є лише однією з кількох доступних і зазвичай використовується для розміщення шаблону введення в одному з декількох класів відповідно до результируючої моделі вихідних даних. Наприклад, якщо введення складається з кодування шаблонів світлого і темного на зображенні рукописних букв, вихідний шар (найвищий на Рисунку) може містити 26 вузлів - по одному для кожної літери алфавіту - для позначення якої букви клас, введений символ від. Це можна було б зробити, виділивши по одному вихідному вузлу для класу та вимагаючи, щоб спрацьовував лише один такий вузол, коли на вході подається шаблон відповідного класу.

Досить немало було сказано про основні конструктивні елементи та їх функціонування. Повертаючись до нашого робочого визначення, зверніть увагу на акцент на вивченні досвіду. У реальних нейронах синаптичні сили можуть, за певних обставин, бути змінені, щоб поведінка кожного нейрона могла змінюватися або пристосовуватися до його конкретного стимулу. У штучних нейронах еквівалентом цього є модифікація вагових значень. Що стосується обробки інформації, тут немає комп'ютерних програм - "знання", якими володіє мережа, повинні зберігатись у своїх вагах, які розвиваються в процесі адаптації до стимулу з безлічі прикладів зразків. В одній навчальній парадигмі, яка називається контрольоване навчання, що використовується разом із мережами типу, показаного на Рисунку 1.3, вхідний шаблон подається в мережу та її реакція, а потім порівнюється з цільовим результатом. З точки зору нашого попереднього прикладу розпізнавання букв, скажімо, може бути введено "А", а мережевий вихід порівняно з класифікаційним кодом А. Тоді різниця між двома моделями виводу визначає, як змінюються ваги. Кожен конкретний рецепт змін становить правило

навчання, деталі якого складають значну частину наступних глав. Коли зроблено необхідні оновлення ваги, представлений інший шаблон, результат порівняння з цільовим показником та внесені нові зміни. Ця послідовність подій повторюється багато разів до тих пір, поки (сподіваємось) поведінка мережі не зійдеться так, що її реакція на кожен шаблон буде близькою до 15 відповідних цілей. Процес в цілому, включаючи будь-яке впорядкування подання шаблону, критерії для завершення процесу тощо, становить алгоритм навчання.

Що станеться, якщо після тренінгу ми представимо мережу з шаблоном, якого вона раніше не бачила? Якщо мережа вивчила основну структуру проблемного домену, то вона повинна класифікувати невидимий зразок правильно, і кажуть, що мережа добре узагальнює. Якщо мережа не має цієї властивості, це трохи більше, ніж таблиця пошуку класифікації навчального набору, і вона мало корисна на практиці. Отже, гарне узагальнення є однією з ключових властивостей нейронних мереж. Нейронні мережі часто використовуються для статистичного аналізу та моделювання даних, в яких їх роль сприймається як альтернатива стандартним методам нелінійної регресії або кластерного аналізу (Cheng & Titterington 1994). Таким чином, вони, як правило, використовуються в задачах, які можуть бути визначені з точки зору класифікації або прогнозування. Деякі приклади включають розпізнавання зображень та мовлення, розпізнавання текстових символів та такі області людського досвіду, як медична діагностика, геологічне дослідження нафти та прогнозування показників фінансового ринку. Цей тип проблем також входить до сфери класичного штучного інтелекту (ШІ), так що інженери та комп'ютерні вчені розглядають нейронні мережі як стиль паралельно розподілених обчислень, забезпечуючи тим самим альтернативу звичайним алгоритмічним методам, які домінують у машинному інтелекті. Це тема, про яку йдеться далі в останньому розділі, але, як коротке пояснення цього терміна зараз, паралелізм відноситься до того факту, що кожен вузол задуманий як функціонуючий незалежно та одночасно (паралельно) з іншими, і "знання" в мережі розподіляються по всьому набору ваг, а не зосереджуються в кількох місцях пам'яті, як у звичайному комп'ютері. Практики в цій галузі не турбуються про біологічний реалізм, і їх часто мотивує простота впровадження рішень у цифровому обладнанні або ефективність та точність певних методів. Хайкін (1994) дає всебічний огляд багатьох нейромережевих методів з інженерної точки зору.

Неврологи та психологи цікавляться мережами як обчислювальними моделями мозку тварин, розробленими шляхом абстрагування тих властивостей справжньої нервової тканини, які є важливими для обробки інформації. Штучні

нейрони, які використовують конекціоністські моделі, часто є надзвичайно спрощеними версіями своїх біологічних аналогів, і багато нейрологів скептично ставляться до остаточної сили цих збіднених моделей, наполягаючи на тому, що для пояснення функції мозку потрібно більше деталей. Тільки час покаже, але, спираючись на знання про те, як реальні нейрони взаємопов'язані як локальні «ланцюги», було зроблено значний прорив у моделюванні функціональності мозку. Хороший вступ до цієї програми обчислювальної нейронауки дають Churchland & Sejnowski (1992). Нарешті, фізиків і математиків цікавить вивчення мереж з-поміж 17 інтересів до нелінійних динамічних систем, статистичної механіки та теорії автоматів. Завдання прикладних математиків - виявляти та формалізувати властивості нових систем за допомогою інструментів, які раніше використовувались в інших галузях науки. Наприклад, існують міцні зв'язки між певним типом сітки та магнітними системами, відомими як спінові окуляри. Повний математичний апарат для вивчення цих зв'язків розроблений (поряд із низкою стислих резюме) Амітом (1989).

Всі ці групи задаються різними питаннями: неврологи хочуть знати, як працює мозок тварин, інженери та інформатики хочуть будувати інтелектуальні машини, а математики хочуть зрозуміти основні властивості мереж як складних систем. Іншу (можливо, найбільшу) групу людей можна знайти в різних промислових і комерційних районах і використовувати нейронні мережі для моделювання та аналізу великих, недостатньо зрозумілих наборів даних, що природним чином виникають на їх робочому місці. Тому важливо розуміти авторську точку зору під час читання літератури. Однак їх спільним фокусним центром є нейронні мережі та потенційно є основою для тісної співпраці. Наприклад, біологи можуть корисно дізнатися у вчених-інформатиків, які обчислення необхідні для вирішення певних проблем тваринами, тоді як інженери можуть використовувати рішення, розроблені природою, щоб їх можна було застосувати в акті "зворотної інженерії".

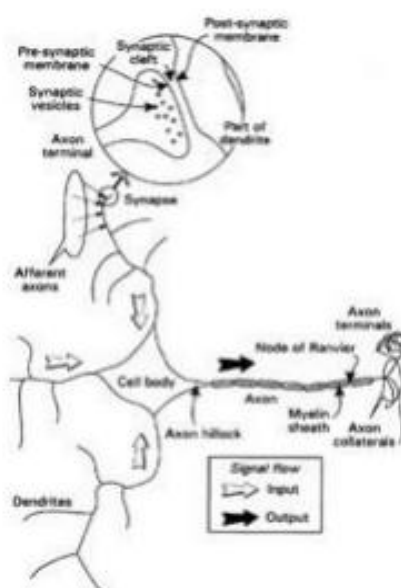
Штучні нейронні мережі можна розглядати як спрощені моделі мереж нейронів, що зустрічаються в природі в мозку тварин. З біологічної точки зору основною вимогою до нейронної мережі є те, що вона повинна намагатись охопити те, що, на нашу думку, є основними особливостями обробки інформації відповідної "реальної" мережі. Для інженера ця відповідність не така важлива, і мережа пропонує альтернативну форму паралельних обчислень, яка може бути більш доречною для вирішення поставленої задачі.

Найпростіший штучний нейрон - це порогова логічна одиниця або TLU. Його основна операція полягає у виконанні зваженої суми своїх вхідних даних, а

потім виведення "1", якщо ця сума перевищує порогове значення, і "0" в іншому випадку. TLU повинен моделювати основну імпульсну нейронну мережу, що складається з імпульсного нейрона з повторюваним механізмом з'єднання реальних нейронів.

Будівельними блоками штучних нейронних мереж є штучні нейрони. Нейрони не тільки надзвичайно складні, але й значно різняться в деталях своєї будови та функції. Тому ми опишемо типові властивості, якими користується більшість нейронів, і зробимо звичайне робоче припущення про коннекціонізм, який забезпечує більшу частину їх обчислювальних здібностей. Читачі, які хочуть дізнатись більше, можуть ознайомитися з одним із багатьох текстів з нейрофізіології; Томпсон (1993) пропонує хороший вступний текст, тоді як більш вичерпні виклади дають Kandel et al. (1991) та Kuffler et al. (1984).

На Рисунку 1.4 показаний стереотипний нейрон, який слід порівняти зі спрощеною діаграмою на Рисунку 1.1. Клітинне тіло або сома містить звичайні субклітинні компоненти або органели, які можна знайти в більшості клітин у всьому тілі (ядро, мітохондрії, тіло Гольджі тощо), але вони не показані на схемі. Натомість ми зосереджуємося на тому, що відрізняє нейрони від інших клітин, дозволяючи нейрону функціонувати як пристрій обробки сигналів. Ця здатність багато в чому зумовлена властивостями поверхневого покриву нейрону або мембрани, яка підтримує широкий спектр електрохімічних процесів. Морфологічно головна відмінність полягає у наборі волокон, що виходять з клітинного тіла. Одне з цих волокон - аксон - відповідає за передачу сигналів до



інших нейронів і, отже, може вважатися вихідним нейроном.

Рисунок 1.4 - Біологічний нейрон.

Всі інші волокна є дендритами, які несуть сигнали від інших нейронів до клітинного тіла, виконуючи тим самим роль нейронних входів. Кожен нейрон має лише один аксон, але може мати багато дендритів. Останні часто мають сильно розгалужену структуру, і тому ми говоримо про дендритні альтанки. Аксон, однак, може розгалужуватися на набір колатералей, що дозволяє встановити контакт з багатьма іншими нейронами. Щодо конкретного нейрона, інші нейрони, що постачають вхід, називаються такими, що відносяться, тоді як аксональний вихід даного нейрона, який розглядається як проекція на інші клітини, називається ефективним.

Кажуть, що різні аксони іннервують певний нейрон і контактують з дендритами в місцях зв'язку, які називаються синапсами. Тут кінець аксона, або термінал аксона, знаходиться в безпосередній близькості від невеликої частини дендритної поверхні - постсинаптичної мембрани. Між синаптичною щілиною є розрив між кінцевою мембраною пресинаптичного аксона та його постсинаптичним аналогом, який має ширину близько 20 нанометрів ( $2 \times 10^{-8}$  м). Для наочності на Рисунку 1.4 показано лише декілька синапсів, але читач повинен уявити їх велику кількість, розташованих по всіх дендритах, а також, можливо, і по тілу клітини. Детальна синаптична структура представлена у схематичному вигляді як вставка на Рисунку.

## 1.2 Справжні і штучні нейрони

Стільки про нервову структуру; як він підтримує обробку сигналу? У рівновазі нервова мембрана працює, щоб підтримувати електричний дисбаланс негативно і позитивно заряджених іонів. Це атоми або молекули, які мають надлишок або дефіцит електронів, де кожен з останніх несе один негативний заряд. Кінцевий результат полягає в тому, що існує потенційна різниця між мембраною, причому внутрішня частина негативно поляризована приблизно на 70 мВ1 щодо зовнішньої. Таким чином, якби ми могли уявити, що застосовуємо вольтметр до мембрани, він зчитував би 70 мВ, причому внутрішня частина була більш негативною, ніж зовні. Головне тут полягає в тому, що нервова мембрана може підтримувати електричні сигнали, якщо її стан поляризації або мембранний потенціал динамічно змінюється. Щоб побачити це, розглянемо випадок поширення сигналу вздовж аксона, як показано на Рисунку 1.5. Сигнали, що поширюються вздовж аксонів, або потенціали дії, мають однакоvu характерну форму, що нагадує гострі імпульсні стрибки. Кожен графік показує знімок



мембранного потенціалу вздовж сегмента аксона, який в даний час передає потенціал єдиної дії, а нижня панель показує ситуацію в якийсь пізніший час щодо верхнього. Іонні механізми, що працюють для створення цього процесу, були вперше розроблені Ходжкіном і Хакслі (1952). Він покладається на взаємодію кожного з іонних струмів через мембрану і його математичний опис є складним.

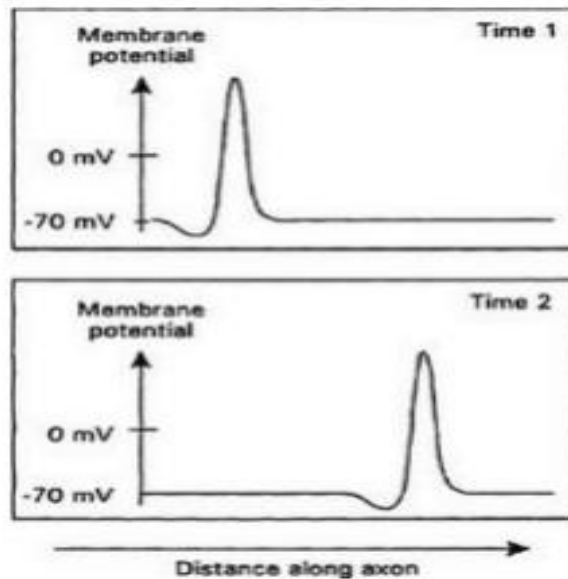


Рисунок 1.5 - Дія потенціалу поширення.

Деталі нас тут не стосуються, але цей приклад служить ілюстрацією того спрощення, яке ми будемо використовувати, коли моделюємо за допомогою штучних нейронів; реальні аксони піддаються складній, нелінійній динаміці, але будуть моделюватися як пасивний вихідний "провід". Багато нейронів мають свої аксони, вкриті жирною речовиною, відомою як мієлін, яка сприяє більш швидкому проведенню потенціалів дії. Він переривається з інтервалом приблизно в 1 мм невеликими немієлінованими сегментами (вузли Ранв'є), які діють як телефонні станції-ретранслятори вздовж телефонного кабелю.

Тепер ми можемо розглянути проходження сигналів через один нейрон, починаючи з потенціалу дії, що досягає афферентного термінатора аксона. Вони містять хімічну речовину або нейромедіатор, що утримується у великій кількості дрібних везикул (буквально «маленькі сфери»). Отримавши потенціал дії, везикули мігрують до пресинаптичної мембрани і вивільняють свій нейромедіатор через синаптичну щілину. Потім передавач хімічно зв'язується з рецепторними ділянками на постсинаптичній мембрані. Це ініціює електрохімічний процес, який змінює стан поляризації локальної мембрани на синапс. Цей постсинаптичний потенціал (PSP) може служити або для деполяризації мембрани з її негативного стану спокою до 0 вольт, або для гіперполяризації мембрани до ще більшого

негативного потенціалу. Як ми побачимо, вироблення нейронних сигналів заохочується деполяризацією, так що позитивні PSP є збудливими PSP (EPSP), тоді як ті, що гіперполяризують мембрану, гальмують (IPSP). Хоча всі потенціали дії мають однаковий характерний профіль сигналу та однакове максимальне значення, PSP можуть приймати безперервний діапазон значень залежно від ефективності синапсу при використанні хімічного передавача для отримання електричного сигналу. PSP поширюється від синапсу, рухається уздовж пов'язаного з ним дендриту до тіла клітини і в решті-решт досягає горбка аксона - початкового сегмента аксона, де він приєднується до соми. Одночасно з цим є тисячі інших синаптичних подій, розподілених по нейрону. Це призводить до безлічі PSP, які постійно надходять до горба аксона, де їх підсумовують для отримання результуючого мембранного потенціалу. Кожна внесена PSP на пагорбі аксона існує протягом тривалого часу (порядку мілісекунд), перш ніж вона в решті-решт занепадає, так що, якщо два PSP приходять трохи не синхронно, вони все одно можуть взаємодіяти в процесі підсумовування. З іншого боку, припустимо, що відбуваються дві синаптичні події, одна близька до іншої, а інша віддалена від соми, завдяки тому, що вони перебувають у кінці довгої дендритної гілки. До того моменту, як PSP з дистального (віддаленого) синапсу досягне горба аксона, той, що походить близько до соми, вже зіпсується. Таким чином, хоча ініціювання PSP може відбуватися синхронно, вони можуть бути неефективними в поєднанні для створення потенціалу дії. Отже, очевидно, що нейрон сумує або інтегрує свої PSP як у просторі, так і в часі. Значні зусилля з моделювання - більшу частину яких започаткував Рейл (1957, 1959) - пішли на опис проведення PSP по дендритах та їх подальшої взаємодії, хоча, як і у випадку з аксонами, коннекціоністські моделі зазвичай трактують їх як пасивні дроти без часових характеристики. Інтегрований PSP на пагорбі аксона впливатиме на його мембранний потенціал, і якщо він перевищує певний поріг (як правило, близько - 50 мВ), генерується потенціал дії, який потім поширюється вниз по аксону, вздовж будь-яких колатералей, в кінцевому підсумку досягаючи терміналів аксона злива синаптичних подій на сусідні нейрони, що знаходяться «нижче за течією» нашої початкової клітини. Насправді "поріг" - це новий феномен, що виникає внаслідок нелінійної природи динаміки Ходжкіна Хакслі, і за певних умов його можна змінити. Однак для багатьох цілей він служить відповідним високорівневим описом того, що насправді відбувається. Після утворення потенціалу дії іонні метаболіти, що використовуються при його виробництві, вичерпуються, і настає короткий рефрактерний період, протягом якого, незалежно від того, яке значення має мембранний потенціал, не може бути ініціації іншого потенціалу дії.

На цьому етапі корисно узагальнити те, що ми до цього часу дізналися про функціональність справжніх нейронів, з огляду на спрощення, необхідне для моделювання їх штучних аналогів.

Сигнали передаються між нейронами за допомогою потенціалів дії, які мають стереотипний профіль і відображають символ "все або нічого"; немає такого поняття, як половина потенціалу дії.

Коли потенціал дії впливає на вхід нейронів (синапс), це ефект PSP, який є змінним або ступінчастим і залежить від фізико-хімічних властивостей синапсу.

PSP можуть бути збудливими або гальмівними.

PSP підсумовуються на пагорбі аксона, а результат виражається як мембранний потенціал

Якщо цей потенціал перевищує порогове значення, ініціюється потенціал дії, який протікає вздовж аксона. Декілька речей тут свідомо опущено. По-перше, вплив, який може мати синаптична структура на значення PSP. Фактори, які тут можуть зіграти певну роль, включають тип та доступність нейромедіатора, постсинаптичні рецептори та синаптичну геометрію. По-друге, просторово-часові взаємозалежності PSP, що виникають в результаті дендритної геометрії, внаслідок чого, наприклад, віддалені один від одного синапси, можуть не ефективно поєднуватися. Нарешті, ми нічого не сказали про динаміку генерації та поширення потенціалу дії. Однак наше резюме послужить відправною точкою для визначення виду штучних нейронів, описаних у цій книзі. Більш біологічно реалістичні моделі покладаються на вирішення динаміки типу Ходжкіна-Хакслі та моделювання дендритів на рівні електричної схеми; подробиці цих методів можна знайти в оглядовій збірці Koch & Segev (1989).

Характер потенціалу дії "все або нічого" може бути охарактеризований за допомогою двозначного сигналу. Такі сигнали часто називають двійковими або булевими і зазвичай приймають значення "0" та "1". Таким чином, якщо у нас є вузол, що приймає  $n$  вхідних сигналів  $x_1, x_2, \dots, x_n$ , то вони можуть приймати лише значення "0" або "1". Відповідно до зауважень попередньої глави, модулюючий ефект кожного синапсу інкапсулюється простим множенням вхідного сигналу зі значенням ваги, де збудливі та гальмівні дії моделюються з використанням позитивних та негативних значень відповідно. Тому ми маємо  $n$  вагових коефіцієнтів  $w_1, w_2, \dots, w_n$  і утворюємо  $n$  продуктів  $w_1x_1, w_2x_2, \dots, w_nx_n$ . Тепер кожен товар є аналогом PSP і може бути негативним або позитивним, залежно від ознаки ваги. Тепер їх слід поєднувати в процесі, який повинен імітувати той процес, що відбувається на пагорбі аксона. Це буде зроблено шляхом

простого їх складання для отримання активації  $a$  (що відповідає потенціалу мембрани аксона-бугра), щоб:

$$a = w_1x_1 + w_2x_2 + \dots + w_nx_n$$

Як приклад, розглянемо одиницю з п'ятьма входами з вагами (0,5, 1,0, -1,0, -0,5, 1,2), тобто  $w_1 = 0,5$ ,  $w_2 = 1,0$ , ...,  $w_5 = 1,2$ , і припустимо, це представлено з входами ( 1, 1, 1, 0, 0) так, що  $x_1 = 1$ ,  $x_2 = 1$ , ...,  $x_5 = 0$ . Використання активації задається:

$$\begin{aligned} a &= (0.5 \times 1) + (1.0 \times 1) + (-1.0 \times 1) + (-0.5 \times 0) + (1.2 \times 0) \\ &= 0.5 \end{aligned}$$

Для емуляції генерування потенціалів дії нам потрібне порогове значення  $\theta$  (грецьке тета), яке, якщо активація перевищує (або дорівнює)  $\theta$ , тоді вузол видає "1" (потенціал дії), а якщо він менше  $\theta$ , тоді він видає "0". Це може бути зображено графічно, як показано на Рисунку 1.6, де вихідним сигналом було позначено символ  $y$ . Це відношення іноді називають ступінчастою функцією або жорстким обмежувачем із зрозумілих причин. Нехай у нашому прикладі  $\theta = 0,2$ ; тоді, оскільки  $a > 0,2$  (пригадайте  $a = 0,5$ ), вихід  $y$  у вузла дорівнює 1. Вся структура вузла показана на Рисунку 1.7, де ваги зображені оточеними знаками множення. Однак, на відміну від Рисунку 1.1, не було зроблено жодних зусиль, щоб показати розмір ваг або сигналів. Цей тип штучного нейрона відомий як порогова логічна одиниця (TLU) і спочатку був запропонований Мак-Каллоком та Піттсом (McCulloch & Pitts 1943).

Більш зручно представляти функціональність TLU у символічній, а не графічній формі. У нас вже є одна форма для активації. Однак це можна записати більш компактно, використовуючи позначення, яке використовує спосіб написання ваг та введених даних.

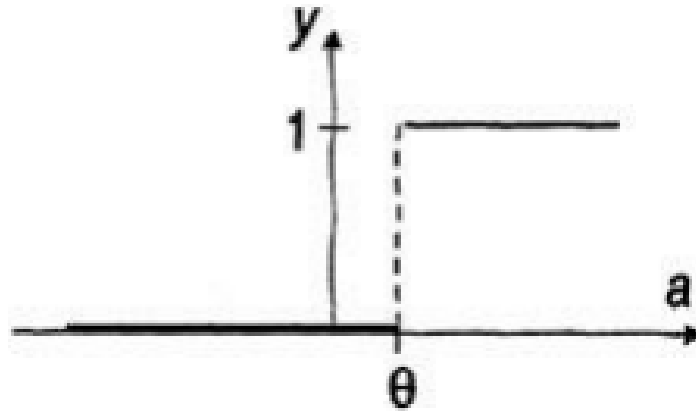


Рисунок 1.6 - Порогове відношення активація-вихід у графічному вигляді.

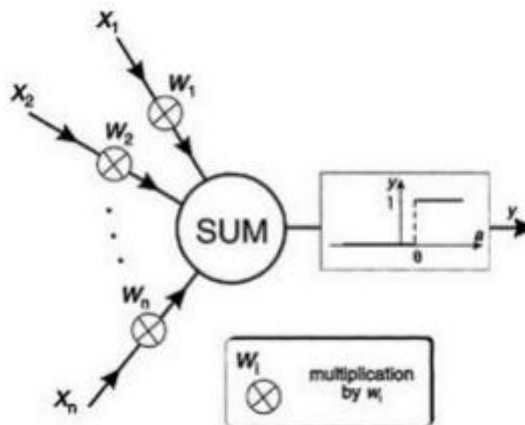


Рисунок 1.7 - TLU.

По-перше, слово про позначення тут є актуальним. Невеликі числа, що використовуються для позначення входів і ваг, називаються індексами. Якби ми писали цифри вгорі (наприклад,  $x_1$ ), вони були б індексами, і, загалом, їх називають індексами незалежно від їх положення. Написавши індекс символічно (а не числово), ми можемо посилатися на величини загально, так що  $x_i$ , наприклад, позначає загальний або  $i$ -й вхід, де передбачається, що  $i$  може бути будь-яким цілим числом від 1 до  $n$ . Подібні зауваження стосуються ваг  $w_i$ . Використовуючи ці ідеї, можна представити їх у більш компактній формі:

$$a = \sum_{i=1}^n w_i x_i$$

де  $\Sigma$  (велика грецька сигма) позначає підсумовування. Вирази над  $i$  і під  $\Sigma$  позначають верхню і нижню межі підсумовування і говорять нам, що індекс  $i$  переходить від 1 до  $n$ . Іноді обмеження опускають, оскільки вони були визначені в іншому місці, і ми просто вказуємо індекс підсумовування (в даному випадку  $i$ ), записуючи його нижче  $\Sigma$ .

Порогове відношення для отримання результату  $y$  може бути записано, як:

$$y = \begin{cases} 1 & \text{if } a \geq \theta \\ 0 & \text{if } a < \theta \end{cases}$$

Зверніть увагу, що в TLU немає згадки про час; одиниця реагує миттєво на свій вхід, тоді як справжні нейрони інтегруються з часом, а також простором. Дендрити представлені (якщо це можна назвати поданням) пасивними сполучними зв'язками між вагами та операцією підсумовування. Генерування потенціалу дії просто представлено пороговою функцією.

Навіть за допомогою цієї простої нейронної моделі ми можемо проілюструвати дві загальні властивості нейронних мереж. Розглянемо дворівневий TLU з вагами (0, 1) і порогом 0,5. Його відповідь на всі чотири можливі набори вводу наведено в Таблиці 1.1.

**Table 2.1** TLU with weights (0, 1) and threshold 0.5.

$x_1$	$x_2$	Activation	Output
0	0	0	0
0	1	1	1
1	0	0	0
1	1	1	1

Таблиця 1.1

Тепер припустимо, що наше обладнання, що реалізує TLU, несправне, так що ваги не утримуються на своїх істинних значеннях і замість цього кодуються як (0,2, 0,8). Переглянута функціональність TLU наведена в таблиці 2.2. Зверніть увагу, що, хоча активація змінилася, вихідний результат такий самий, як і вихідний TLU. Це пов'язано з тим, що зміни в активації, доки вони не переступають поріг, не спричиняють жодних змін у виведенні. Таким чином, пороговій функції не важливо, чи буде активація трохи нижче  $\theta$ , чи вона набагато менша ніж  $\theta$ ; він все одно видає 0. Аналогічним чином, неважливо, наскільки активація перевищує  $\theta$ , TLU завжди подає 1 як вихід.

Таблиця 1.2

**Table 2.2** TLU with weights (0.2, 0.8) and threshold 0.5.

$x_1$	$x_2$	Activation	Output
0	0	0	0
0	1	0.8	1
1	0	0.2	0
1	1	1	1

Така поведінка характерна для нелінійних систем. У лінійній системі випуск пропорційно пов'язаний із входом: малі / великі зміни на вході завжди спричиняють відповідні малі / великі зміни на виході. З іншого боку, нелінійні відносини не піддаються обмеженню пропорційності, тому величина зміни обсягу виробництва не обов'язково відображає величину вхідних даних. Таким чином, у нашому прикладі TLU активація може змінюватися від 0 до 0,2 (різниця 0,2) і не матиме різниці у вихідних даних. Якщо, однак, це змінилося б з 0,49 до 0,51 (різниця 0,02), вихід раптово змінився б від 0 до 1.

З усього цього ми робимо висновок, що TLU надійні за наявності відмови обладнання;

якщо наше обладнання "злегка виходить з ладу", TLU все ще може функціонувати чудово, як результат його нелінійної функціональності.

Таблиця 1.3

**Table 2.3 TLU with degraded signal input.**

$x_1$	$x_2$	Activation	Output
0.2	0.2	0.2	0
0.2	0.8	0.8	1
0.8	0.2	0.2	0
0.8	0.8	0.8	1

Припустимо тепер, що замість того, щоб змінити ваги, вхідні сигнали дещо погіршились, наприклад, через шум або часткову втрату потужності, так що те, що раніше було "1", тепер позначається 0,8, а "0" стає 0,2. Отримана функція TLU показана в таблиці 2.3. Ще раз отримана функція TLU однакова, і застосовується подібне міркування, яке включає нелінійність, передбачену порогом. Висновок полягає в тому, що TLU надійний при наявності шумних або пошкоджених входів сигналу. Читачеві пропонується вивчити випадок, коли і ваги, і сигнали були погіршені способом, зазначеним тут. Звичайно, якщо ми збільшимо величину, на яку ваги або сигнали були занадто сильно змінені, TLU врешті-решт буде неправильно реагувати. У великій мережі, зі збільшенням ступеня погіршення апаратного забезпечення та / або сигналу, кількість одиниць TLU, що дають неправильні результати, також поступово зростатиме. Цей процес називається "витонченою деградацією", і його слід порівнювати з тим, що відбувається в звичайних комп'ютерах, коли зміна одного компонента або втрата потужності сигналу вздовж однієї доріжки друкованої плати може призвести до повного виходу машини з ладу.

### 1.3 Небінарний сигнальний зв'язок

Сигнали, з якими розглядалися дотепер (як для реальних, так і для штучних нейронів), набули лише двох значень. У разі справжніх нейронів це напруга стрибка потенціалу дії та потенціал спокою мембрани аксона. Для TLU вони були зручно позначені як "1" та "0" відповідно. Однак справжні нейрони кодують свої значення сигналів у шаблонах дії потенціалу дії, а не просто наявністю або відсутністю одного такого імпульсу. Спостерігається багато характерних закономірностей (Connors & Gutnick 1990), два загальних приклади показані на Рисунок 1.8.

Частина (а) показує безперервний потік стрибків потенціалу дії, тоді як (б) показує візерунок, за яким після серії імпульсів настає період спокою, причому ця послідовність повторюється необмежено довго.

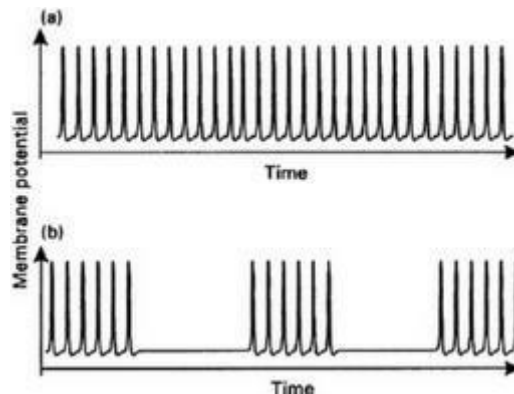


Рисунок 1.8 - Нейронні схеми випалу.

Безперервний потік, як у (а), може характеризуватися частотою появи потенціалу дії в імпульсах в секунду, і спокусливо припустити, що це, власне, код, який сигналізує нейрон. Це переконливо продемонстрував Хартлайн (1934, 1940) для зорових нейронів підкованого краба *Limulus*, в якому він показав, що швидкість стрільби зростала із збільшенням інтенсивності зорового подразника. Хоча доступно багато нейронних кодів (Bullock et al. 1977), частотний код, здається, використовується в багатьох випадках. Якщо  $f$  - частота нейронного випалу, тоді ми знаємо, що  $f$  обмежується знизу нулем, а вище деяким максимальним значенням  $f_{max}$ , яке регулюється тривалістю періоду вогнетривкого впливу. Зараз існує два способи кодування  $f$  у наших штучних нейронах. По-перше, ми можемо просто розширити подання сигналу до безперервного діапазону і безпосередньо представити  $f$  як вихід нашої одиниці. Такі сигнали, безперечно, можна обробляти на вході TLU, як ми зауважували під час вивчення наслідків погіршення сигналу. Однак використання ступінчастої



функції на виході обмежує двійкові сигнали, так що, коли TLU підключені до мереж (і вони працюють належним чином), відсутня можливість постійно градуйованих сигналів. Це можна подолати, "пом'якшивши" функцію кроку до безперервної функції "стиснення", щоб вихід у плавно залежав від активації  $a$ . Однією зручною формою для цього є логістична сигмоїда (або іноді просто "сигмоїдна"), показана на Рисунку 1.9.

Оскільки тенденція до великих позитивних значень, сигмоїда прагне до 1, але насправді ніколи не досягає цього значення. Подібним чином він наближається - але ніколи не досягає - 0, оскільки має тенденцію до великих негативних значень. Немає значення, що верхня межа не є  $f_{max}$ , оскільки ми можемо просто помножити значення сигмоїди на  $f_{max}$ , якщо хочемо інтерпретувати  $y$  як реальну швидкість стрільби. Сигмоїдна симетрична відносно значення осі  $y$  0,5; відповідне значення активації може розглядатися як переосмислення порогу і позначається  $\theta$ .

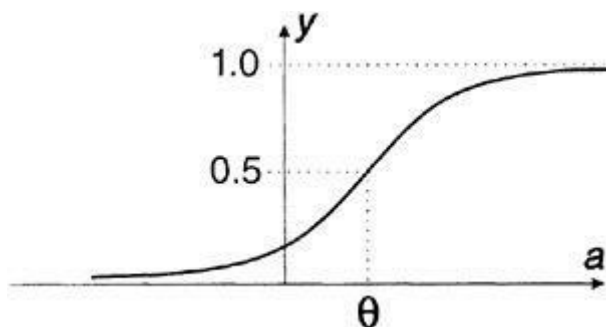


Рисунок 1.9 - Приклад функції стиснення — сигмоїда.

Сигмоїдна функція умовно позначається грецькою малою буквою сигма,  $\sigma$ , і знаходить математичний вираз відповідно до співвідношення

$$y = \sigma(a) \equiv \frac{1}{1 + e^{-(a-\theta)/\rho}}$$

де  $e$  (число Ейлера) приблизно дорівнює 2.7183 - математична константа, яка, як і число  $\rho$ , має нескінченне десяткове розширення. Величина  $\rho$  (грецьке rho) визначає форму функції, великі значення роблять криву більш рівною, тоді як малі значення роблять криву більш крутим. У багатьох текстах цей параметр опущений, так що йому неявно присвоюється значення 1. Завдяки зменшенню  $\rho$  ми отримуємо функції, які виглядають все ближче до жорсткого обмежувача, що використовується в TLU, так що вихідна функція останнього може розглядатися як особливий випадок. Посилання на  $\theta$  як поріг тоді стає більш правдоподібним, оскільки воно бере на себе роль того самого параметра в TLU. Штучні нейрони або одиниці, які використовують сигмоїдальне відношення

виходу, називаються напівлінійним типом. Активація все ще задається рівнянням (2.2), але тепер вихідний результат задається (2.4). Вони становлять основу великої роботи в нейронних мережах, оскільки плавна вихідна функція полегшує їх математичний опис. Термін "напівлінійний" походить від того факту, що ми можемо апроксимувати сигмоїду за допомогою неперервної кусково-лінійної функції, як показано на Рисунку 1.10. У значній області інтересу при проміжних значеннях активації вихідна функція являє собою лінійне відношення з ненульовим нахилом.

Як альтернативу використанню безперервних або аналогових значень сигналу, ми можемо емулювати реальний нейрон і кодувати сигнал як частоту появи «1» в імпульсному потоці, як показано на Рисунку 1.11.

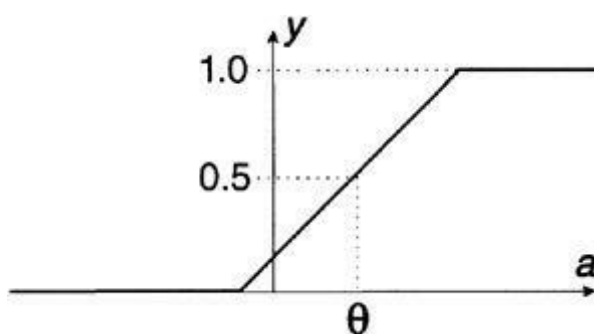


Рисунок 1.10 - Частично-лінійне наближення сигмоїди.

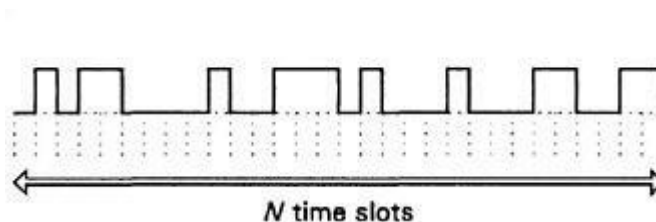


Рисунок 1.11 Потік вихідних імпульсів від стохастичного вузла.

Час ділиться на дискретні "слоти", і кожен слот заповнюється або 0 (без імпульсу), або 1 (імпульс). Вихід одиниці формується точно так само, як і раніше, але, замість того, щоб безпосередньо відправляти значення сигмовидної функції, ми інтерпретуємо це як ймовірність випромінювання імпульсу або "1". Процеси, що регулюються ймовірнісними законами, називаються стохастичними, щоб ці вузли могли називати стохастичними напівлінійними одиницями, і вони в цілому видають досить близькі сигнали, досить близькі за загальним виглядом до справжніх нейронів. Як одиниці нижче за течією, які приймають ці сигнали, повинні інтерпретувати свої входи? Тепер вони повинні інтегрувати деяку кількість,  $N$ , часових інтервалів. Отже, припустимо, що аферентний вузол генерує імпульси з ймовірністю  $y$ . Очікуване значення кількості імпульсів за цей час дорівнює  $yN$ , але, загалом, фактично вироблене число,  $N_1$ , не обов'язково буде

дорівнювати цьому. Найкращою оцінкою, яку може зробити вузол, що приймає ці сигнали, є частка  $N1 / N$  протягом часу інтеграції. Ситуація така в експерименті підкидання монет. Основна ймовірність отримання "голови" становить 0,5, але в будь-якій конкретній послідовності підкидань кількість голів  $Nh$  не обов'язково становить половину від загальної кількості. Однак із збільшенням числа кидок  $N$  частка  $Nh / N$  з часом наближається до 0,5.

Спосіб моделювання тимчасового підсумовування PSP на пагорбі аксона полягає у використанні швидкості зміни активації як основної визначальної величини, а не самої активації. Повне лікування вимагає використання розділу математики, відомого як числення, але отримана поведінка може бути описана досить прямолінійно. Однак ми приймемо позначення числення  $dx / dt$  для швидкості зміни величини  $x$ . Не можна переоцінити, що це слід читати як єдину символічну сутність " $dx / dt$ ", а не як  $dx$ , поділену на  $dt$ . Щоб уникнути плутанини з попереднім позначенням, необхідно ввести інший символ для зваженої суми входів, тому ми визначимо

$$s = \sum_{i=1}^n w_i x_i$$

Потім швидкість зміни активації,  $da / dt$ , визначається як:

$$\frac{da}{dt} = -\alpha a + \beta s$$

де  $\alpha$  (альфа) і  $\beta$  (бета) є позитивними константами. Перший член призводить до спаду активації, тоді як другий представляє вхідні дані від інших одиниць. Як зазвичай вихід  $y$  задається сигмоїдою активації,  $y = \sigma(a)$ . Така одиниця іноді називається негерметичним інтегратором з причин, які стануть очевидними незабаром.

Існує точний фізичний аналог негерметичного інтегратора, з яким ми всі знайомі. Розглянемо резервуар для води, який має вузький вихід біля основи і який також подається шлангом або краном, як показано на Рисунку 1.12 (ми можемо подумати про ванну з меншим дренажним отвором, ніж зазвичай). Нехай швидкість, з якою вода протікає через шланг, становить  $s$  літрів на хвилину, а глибина води повинна бути  $a$ . Якби розетку було заблоковано, швидкість зміни рівня води була б пропорційною  $s$ , або  $da / dt = \beta s$ , де  $\beta$  - константа. Тепер припустимо, що припливу немає, але вихід працює. Швидкість виходу води прямо пропорційна тиску води на виході, який, у свою чергу, пропорційний глибині води  $\alpha$  в резервуарі. Таким чином, швидкість викидів води може бути записана як  $\alpha$  літрів на хвилину, де  $\alpha$  деяка константа. Рівень води зараз знижується, так що швидкість її зміни зараз від'ємна, і ми маємо  $da / dt = -\alpha$ . Якщо і шланг, і

випускний отвір функціонують, тоді  $da / dt$  - це сума внесків від обох, і його регулююче рівняння точно таке ж, як і для нервової активації. Під час подальшої дискусії, можливо, варто повернутися до цієї аналогії, якщо читач сумнівається у тому, що відбувається.

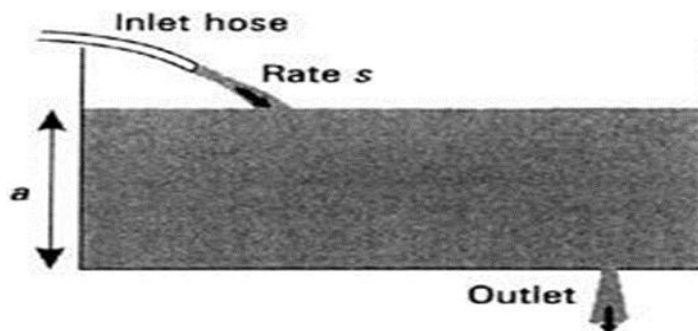


Рисунок 1.12 - Аналогія резервуару для води для негерметичних інтеграторів.

Повертаючись до нейронної моделі, активація може бути негативною або позитивною (тоді як рівень води в резервуарі завжди позитивний). Таким чином, якщо поставити  $s = 0$ , щоб пристрій не мав зовнішнього входу, є два випадки:

(a)  $a > 0$ . Тоді  $da / dt < 0$ . Тобто швидкість змін є негативною, що означає зменшення з часом.

(b)  $a < 0$ . Тоді  $da / dt > 0$ . Тобто швидкість змін є позитивною, що означає збільшення з часом.

Це проілюстровано на Рисунку 1.13, де ліва та права сторони відповідають випадкам (a) та (b) відповідно. В обох випадках активність поступово наближається до нульового значення спокою. Саме цей процес розпаду веде до «негерметичної» частини назви підрозділу. У TLU або напівлінійному вузлі, якщо ми відкликаємо введення, активність відразу стає нульовою. Однак у новій моделі пристрій має своєрідну короткочасну пам'ять попереднього вводу до того, як його було вилучено. Таким чином, якщо це було негативним, активація залишається негативним протягом деякого часу, при цьому зберігається відповідна умова для нещодавно знятого позитивного входу.

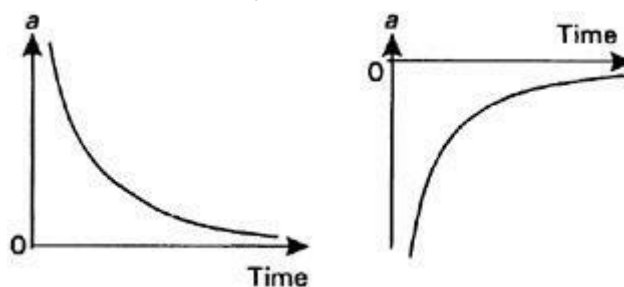


Рисунок 1.13 - Розпад активації в негерметичному інтеграторі.

Припустимо тепер, що ми починаємо з активації нуль і відсутність вводу, і подаємо постійний вхід  $s = 1$  протягом часу  $t$  перед тим, як зняти його знову. В результаті цього активація показана на Рисунку 1.14. Активація починає зростати, але робить це досить мляво. Після того як  $s$  зведено до нуля, розпад відбувається способом, описаним вище. Якби  $s$  підтримувалося досить довго, то а зрештою досягло б постійного значення. Щоб побачити, що це, покладемо  $da / dt = 0$ , оскільки це твердження про відсутність швидкості зміни  $a$ , а  $a$  постійна при деякому рівноважному значенні  $a_{eqm}$ . Поставивши  $da / dt = 0$  у задає

$$a_{eqm} = \left(\frac{\beta}{\alpha}\right) s$$

тобто постійна частка  $s$ . Якщо  $\alpha = \beta$ , то  $a_{eqm} = s$ . Швидкість, з якою  $a$  може реагувати на вхідну зміну, може характеризуватися часом, необхідним для досягнення деякої частки  $a_{eqm}$  ( $0,75a_{eqm}$ , скажімо), і називається часом наростання.

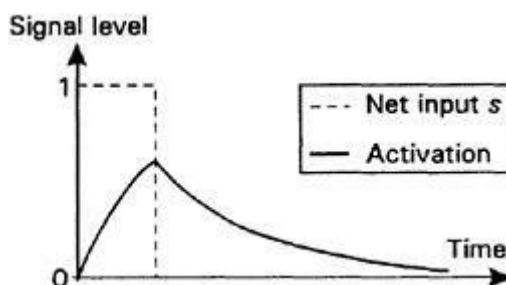


Рисунок 1.14 Вхідний імпульс до негерметичного інтегратора.

Припустимо тепер, що наступний вхідний імпульс подається незабаром після того, як перший був знятий. Нова поведінка показана на Рисунку 1.15. Тепер активація починає прийматися знову, коли подається другий вхідний сигнал, і оскільки тимчасовий проміжок часу не встиг спасти до значення спокою, пікове значення, отримане цього разу, більше, ніж раніше. Таким чином, два сигнали взаємодіють між собою, і відбувається тимчасове підсумовування або інтегрування (частина "інтегратора" назви блоку). У TLU активація, звичайно, буде просто дорівнює  $s$ . Значення констант регулюють швидкість занепаду та час наростання відповідно, і в міру їх збільшення швидкість занепаду збільшується, а час наростання падає. Зберігання  $\alpha = \beta$  і дозволяючи обом стати дуже великими, отже, дозволяє дуже швидко підніматися і опускатися і досягти рівноваги при  $s$ . У міру збільшення цих констант подальша поведінка  $a$  стає невідмінною від поведінки TLU, що, отже, можна розглядати як приватний випадок негерметичного інтегратора з дуже великою постійною  $\alpha$ ,  $\beta$  (і, звичайно, дуже крутою сигмоїдою).

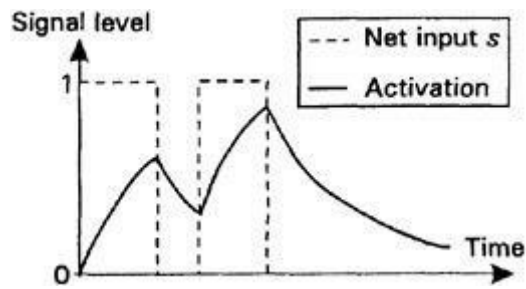


Рисунок 1.15 - Активация негерметичного интегратора (суцільна лінія) для двох квадратних вхідних імпульсів (пунктирна лінія).

Отже, Функція справжніх нейронів надзвичайно складна. Однак основні атрибути обробки інформації можуть бути зведені наступним чином. Нейрон отримує вхідні сигнали від багатьох інших (аферентних) нейронів. Кожен такий сигнал модулюється (за допомогою синаптичного механізму) від стрибка напруги потенціалу дії в постійно змінний (градуований) постсинаптичний потенціал (PSP). PSP інтегруються дендритними альтанками як у просторі (багато синаптичних входів), так і в часі (PSP не відпадають до нуля миттєво). PSP можуть бути збудливими або гальмівними, і їх інтегрованим результатом є зміна мембранного потенціалу на пагорбі аксона, що може служити для деполяризації (збудження або активації) або гіперполяризації (гальмування) нейрона. Динаміка мембрани при цих змінах є складною, але її можна описати в багатьох випадках, припускаючи, що існує мембранний поріг потенціалу, за яким генерується потенціал дії і нижче якого не відбувається такої події. Шлях потенціалів дії становить нейронний "вихід". Вони віддаляються від тіла клітини вздовж аксона, поки не досягають терміналів аксонів (в синапсах), на яких цикл подій знову ініціюється. Інформація багато в чому кодується в нейронах, але загальним методом є використання частоти або швидкості вироблення потенціалів дії.

Інтегрування сигналів у просторі може моделюватися за допомогою лінійно зваженої суми входів. Тоді вважається, що синаптична дія еквівалентна множенню на вагу. TLU моделює потенціал дії за допомогою простого порогового механізму, який дозволяє два рівні сигналу (0 або 1). Швидкість стрільби може бути представлена безпосередньо в напівлінійному вузлі, дозволяючи безперервний вихід або (у стохастичному варіанті), використовуючи це значення як ймовірність для створення сигнальних імпульсів. Інтеграція з часом забезпечується моделлю негерметичного інтегратора. Всі штучні нейрони демонструють стійку поведінку при погіршенні вхідних сигналів та апаратних збоях.

## РОЗДІЛ 2 ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ ЗАХВОРЮВАНЬ COVID-19

### 2.1 Перші спроби створення нейронної мережи для дослідження пандемії

Понад 79,2 мільйона підтверджених випадків COVID-19 та 1,7 мільйона смертей були спричинені ГРВІ-CoV-2; захворювання було названо COVID-19 Всесвітньою організацією охорони здоров'я. Контроль над епідемією COVID-19 став найважливішим питанням у всьому світі, але є обмежені дослідження, які досліджують глобальну тенденцію пандемії COVID-19 разом із політичними заходами кожної країни.

З'явилася потреба розробити мережеву систему штучного інтелекту (ШІ) для аналізу динамічної тенденції пандемії COVID-19, полегшення прогнозування та прогнозного моделювання та створення візуалізації теплових карт політичних заходів у 171 країні.

Для аналізу та дослідження COVID-19 була створена система AI пандемії COVID-19 (CPAIS), яка інтегрувала два набори даних: набір даних від Оксфордського урядового відстежувача COVID-19 від Школи управління Блаватник, який підтримується Оксфордським університетом, та набір даних COVID-19 сховище даних, яке було створене Центром системної науки та техніки університету Джона Гопкінса. У цьому дослідженні використовувались чотири статистичні методи та методи глибокого навчання для прогнозування: авторегресивна інтегрована ковзна середня (ARIMA), нейромережа прямого пересилання (FNN), нейронна мережа багат шарового персептронну (MLP) та довга короткочасна пам'ять (LSTM). Що стосується 1-річних записів (тобто даних цілих часових рядів), записи за останні 14 днів служили набором перевірки для оцінки ефективності прогнозу, тоді як попередні записи служили набором навчальних занять.

Загалом до онлайн-системи було включено 171 країну, яка фігурувала в обох базах даних. CPAIS був розроблений для вивчення варіацій, тенденцій та прогнозів, пов'язаних з пандемією COVID-19 у кількох країнах. Наприклад, кількість підтверджених щомісячних випадків у Сполучених Штатах досягла локального піку в липні 2020 року та ще одного піку - 6 368 591 у грудні 2020

року. Динамічна теплова карта з політичними заходами відображає зміни в заходах COVID-19 для кожної країни. Загалом 19 заходів було включено в три розділи, представлені на веб-сайті, і лише 4 із 19 заходів були постійними заходами, пов'язаними з фінансовою підтримкою або інвестиціями. Для увімкнення прогнозування COVID-19 використовувались моделі глибокого навчання; ефективність ARIMA, FNN та нейромережі MLP не була стабільною, оскільки їх точність прогнозу була лише кращою, ніж LSTM для кількох країн. LSTM продемонстрував найкращу точність прогнозу для Канади, оскільки середньоквадратична помилка (RMSE), середня абсолютна похибка (MAE) та середня абсолютна похибка у відсотках (MAPE) становили 2272,551, 1501,248 та 0,2723075, відповідно. ARIMA (RMSE = 317,53169; MAPE = 0,4641688) та FNN (RMSE = 181,29894; MAPE = 0,2708482) продемонстрували кращі показники для Південної Кореї.

CPAIS збирає та узагальнює інформацію про пандемію COVID-19 та пропонує візуалізацію даних та прогнозування на основі глибокого навчання. Це може бути корисним посиланням для прогнозування серйозних спалахів або епідемій. Більше того, система щодня оновлюється та включає останню інформацію про вакцинацію, яка може змінити динаміку пандемії.

У грудні 2019 року про перші випадки нового респіраторного захворювання, спричиненого новим коронавірусом, було зареєстровано в Ухані, провінція Хубей, Китай [1]. Згодом новий коронавірус був ідентифікований і названий SARS-CoV-2, а захворювання, спричинене SARS-CoV-2, було названо COVID-19 Всесвітньою організацією охорони здоров'я (ВООЗ). З тих пір, як були зареєстровані перші випадки, багато підтверджених випадків повідомлялося в різних інших країнах. До 11 березня 2020 року було зареєстровано понад 118 000 підтверджених випадків та 4291 смерть у 114 країнах. ВООЗ оголосила спалах COVID-19 пандемією, яка продовжує погіршуватися. Станом на 27 грудня 2020 року було зареєстровано понад 79,2 мільйона підтверджених випадків та 1,7 мільйона смертей. Управління COVID-19 постало актуальною глобальною проблемою. Багато досліджень досліджували фактори, що сприяють поширенню COVID-19. Демографічні, географічні та економічні фактори вплинули на поширення хвороби. Однак соціальні фактори, особливо реакція уряду на пандемію, суттєво вплинули на тяжкість захворювання в певних країнах. Деякі країни продемонстрували, що впровадження суворих стратегій управління охороною здоров'я може успішно контролювати поширення інфекції та підтримувати нормальне функціонування суспільства.



Швидкий розвиток штучного інтелекту (ШІ) у галузі охорони здоров'я відкриває нові можливості для дослідників медицини. Є багато досліджень, які використовують методи ШІ для прогнозування захворювань, такі як Yu та співавт., Які створили Інтернет-систему оцінки стану здоров'я для метаболічного синдрому та хронічних захворювань нирок. Лін та інші використали багатоцентрові дані для розробки системи оцінки смертності від прогнозу смертності від кінцевої стадії. Айюбаде та ін. Проаналізували рівень захворюваності на COVID-19 в Ірані за допомогою даних Google Trends та методів глибокого навчання. Йонг та співавтори об'єднали декілька онлайн-даних COVID-19 для підготовки та оцінки п'яти моделей машинного навчання, що не мають часового ряду, для прогнозування підтвердженого зростання інфекції. Ці дослідження показали, що ШІ підходить для оцінки тенденцій захворювання та може надавати урядам інформацію, яка може бути використана для запобігання розповсюдженню. Існує велика кількість досліджень щодо прогнозування ШІ, пов'язаного з COVID-19, та використання даних мобільних датчиків з клітинною трансляцією для виявлення та управління потенційними контактами.

Однак більшість цих досліджень проводились у певному регіоні чи окремій країні. Існує консенсус у галузі охорони здоров'я щодо того, що вакцинація є ефективною стратегією профілактики. Однак, щодо його ефективності та витрат на лікування, необхідне тривале подальше розслідування для оцінки клінічних ефектів вакцин, які не пройшли стандартний процес затвердження, та тестів їх середньо- та довгострокових побічних ефектів на різні групи. Більше того, різні дослідження зосереджували увагу на різних часових рамках у прогнозуванні тенденцій пандемії. Вони зробили однаковий висновок: існує велика ймовірність того, що COVID-19 залишиться поширеною хворобою або стане ендеміком у майбутньому, і ми повинні навчитися співіснувати з нею. Багато факторів впливають на розвиток пандемії (наприклад, імунітет стада), а реакції уряду та окремих людей різняться в різних країнах. Успішні заходи профілактики та боротьби з епідеміями залишаються найефективнішим рішенням проблем громадського здоров'я. Однак існує обмежена література про взаємозв'язок урядових реакцій та тяжкості поширення COVID-19 на національному рівні.

Таким чином, було створено Інтернет-систему ШІ, яка містить світові дані, пов'язані з COVID-19, урядовими реакціями кожної країни на пандемію COVID-19 та даними про населення кожної країни. Система ШІ пандемії COVID-19 (CRAIS) може бути використана для аналізу динамічної тенденції пандемії COVID-19, полегшення прогнозування та прогнозного моделювання та створення візуалізації теплових карт політичних заходів у різних країнах.

CPAIS інтегрував два набори даних: набір даних з Оксфордського урядового відстежувача COVID-19 (OxCGRT) Школи уряду Блаватника, який підтримується Оксфордським університетом, і набір даних із сховища даних COVID-19 який був створений Центром системної науки та техніки університету Джона Хопкінса (CSSE). Репозитарій даних COVID-19 також містить дані про населення кожної країни, які отримані зі Світових перспектив народонаселення ООН. Загалом до системи було включено 171 країну, яка фігурувала в базах даних.

CPAIS було поміщено на сервер і вбудовано в моделі глибокого навчання з часовими рядами, щоб забезпечити аналіз прогнозування за допомогою статистичної програми R, версія 3.6.3 (The R Foundation). Для цього використовували фреймворк React.js, версія 16.14.0; мову стилів Sass (Синтаксично дивовижні таблиці стилів), версія 4; та мову програмування JavaScript ES6 для інтерфейсної реалізації. Що стосується внутрішньої реалізації, то там використовували Java 8; Spring Boot, версія 2.0.2 (VMware, Inc); і R як мови програмування, і також використовували MySQL (Structured Query Language), версія 5.7.21, базу даних як систему зберігання. Крім того, ця система на основі ШІ запрограмована на оновлення, шляхом автоматичного отримання інформації з усіх наборів даних щоранку о 9 ранку (GMT + 8). Автоматичне отримання може бути узагальнено на таких трьох кроках:

- Налаштування сканера для отримання даних із вихідних баз даних;
- Інтеграція оновлених даних у нашу власну базу даних MySQL;
- Проведення статистичного аналізу за допомогою процедура, що зберігається в базі даних.

Сховище даних COVID-19, створене CSSE Університету Джона Хопкінса, містить три категорії даних щодо захворюваності на COVID-19 - підтверджених випадків, випадків відновлення та кількості смертей - із геолокацією країн, отриманою з 192 постраждалих країн з 21 січня 2020 року. Для більшості з країн доступні дані на рівні країн щодо кількості зареєстрованих випадків. Для деяких країн доступні дані про рівні провінцій та міст щодо зареєстрованих випадків. Щоб всебічно зобразити пандемію COVID-19, ці дані були архивовані на рівні країн. Кількість зареєстрованих випадків оновлювалася щодня, використовуючи дані, отримані з багатьох Інтернет-джерел. Кількість випадків було отримано в ВООЗ та регіональних та місцевих управліннях охорони здоров'я постраждалих країн, включаючи їх центри з контролю та профілактики захворювань. Всі дані передавалися вільно через GitHub.

OxCGRT збирає та документує відповіді уряду на пандемію COVID-19 на основі кількох параметрів з 1 січня 2020 р. Набір даних включає 183 країни та 20

пунктів (19 показників та 1 безкоштовну відповідь), що характеризують відповіді уряду. Існує три типи елементів:

- Порядкова шкала ступеня важкості або інтенсивності;
- Числова шкала для конкретних чисел;
- Текст для інших типів інформації.

Далі ці елементи можна класифікувати на чотири групи:

- Політика стримування та закриття (8 показників);
- Економічна політика (4 показники);
- Політика системи охорони здоров'я (7 показників);
- Різноманітна політика ( 1 безкоштовна відповідь).

Різні правила не були включені в цю систему, оскільки вони оцінювались за допомогою формату вільного тексту та мали обмежені дані. Дані OxCGRT отримували із загальнодоступних джерел та регулярно оновлювались на GitHub.

Для цього дослідження було розглянуто чотири моделі часових рядів. Кожна модель була застосована до всіх країн нашої системи для полегшення прогнозування. Що стосується 1-річних записів (тобто даних цілих часових рядів), записи за останні 14 днів служили набором перевірки, тоді як попередні записи служили набором навчальних занять. Використовуючи записи за останні 14 днів, результати прогнозування оцінювали на основі наступних п'яти індексів: середня помилка (ME), середньоквадратична помилка (RMSE), середня абсолютна помилка (MAE), середня відсоткова помилка (MPE) та середня абсолютна процентна помилка (MAPE). RMSE, MAE та MAPE завжди є позитивними значеннями, тоді як RMSE, MPE та MAPE - це масштабовані міри. Гіперпараметри для кожної моделі можна знайти в таблиці S1.

Мультимедійний додаток 1 та діаграму нейронних мереж можна знайти на Рисунку 2.1. R, версія 3.6.3 (The R Foundation), була використана для проведення статистичного аналізу та застосування методів глибокого навчання.

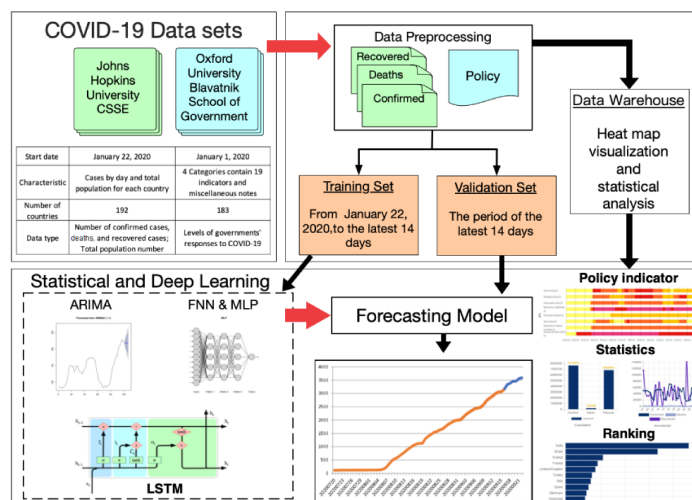


Рисунок 2.1 - Структура системи ШІ пандемії COVID-19 (CPAIS). ARIMA: авторегресивна інтегрована ковзна середня; CSSE: Центр системної науки та інженерії; FNN: нейронна мережа прямого зв'язку; LSTM: довга короткочасна пам'ять; MLP: багатошаровий перцептрон; NN: нейронна мережа.

Модель авторегресійної інтегрованої ковзної середньої (ARIMA) - це статистичний регресійний аналіз, який використовує дані часових рядів, щоб або краще зрозуміти набір даних, або передбачити майбутні тенденції. Метою ARIMA є прогнозування майбутніх тенденцій шляхом вивчення відмінностей між значеннями в серії, а не за допомогою фактичних значень. Три основні компоненти ARIMA - це авторегресія, інтеграція та ковзаюче середнє. Авторегресія відноситься до моделі зі змінною змінною, яка регресує за своїми значеннями відставання. Інтеграція представляє різницю між значеннями даних та їх попередніми значеннями для стаціонарних часових рядів. Ковзне середнє включає залежність між спостереженням і терміном помилки від моделі ковзного середнього. Модель ARIMA можна зрозуміти, окресливши кожен компонент, який служить параметром зі стандартними позначеннями. Для моделей ARIMA існує три стандартні позначення, де цілі значення служать заміною параметрів, щоб вказати тип використовуваної моделі ARIMA.

Параметри можна визначити наступним чином:

p: кількість затримок часу

d: ступінь диференціації

q: розмір вікна ковзного середнього.

У цьому дослідженні було використано функцію `auto.arima` для R, яка повертає найкращу модель ARIMA, що базується або на значенні інформаційного критерію Акаїке (AIC), або на байєсівському значенні інформаційного критерію. Функція шукає можливі моделі в рамках обмежень порядку, передбачених у пакеті прогнозів для R.

Нейронна мережа прямого зв'язку (FNN) - це найпростіший тип штучної нейронної мережі. Алгоритм FNN має біологічний натхнення. Він складається з кількох простих нейроподібних одиниць, які організовані шарами. У FNN інформація рухається в одному напрямку - від вхідних вузлів, через приховані вузли та до вихідних вузлів. Механізм FNN відрізняється від механізму повторюваних нейронних мереж (RNN) тим, що зв'язки між блоками не утворюють циклів або циклів у FNN [38,39]. У цьому дослідженні було використано функцію `nnetar` для R, яка створює FNN з єдиним прихованим шаром та відсталими входами з метою прогнозування одновимірних часових рядів. Крім

того, у пакеті прогнозування функція вписується в єдину приховану нейронну мережу для прогнозування, з функцією `nnet`, включеною в пакет `nnet` для R.

Багатошарова нейромережа Перцептрон (MLP), як і FNN, є загальноприйнятими мережами зворотного зв'язку для глибокого навчання. Нейронна мережа MLP - це також керований алгоритм навчання, що використовується для класифікації. Основна відмінність полягає в тому, що між вхідним і вихідним рівнем може бути кілька нелінійних шарів, які називаються прихованими шарами, які є справжнім обчислювальним рушієм нейронної мережі MLP. Нейромережі MLP використовують для навчання техніку навчання, звану зворотним розповсюдженням. Їх багатошаровість та нелінійна активація відрізняють нейромережі MLP від лінійного перцептрона. Іншими словами, нейромережі MLP призначені для вирішення нелінійно відокремлюваних проблем. Зокрема, блоки нейронних мереж MLP застосовують сигмоподібну функцію як функцію активації. У техніці зворотного розповсюдження різниця між вихідними значеннями та відповіддю на основну істину обчислюється за допомогою попередньо визначених функцій помилки. Помилка передається назад через мережу. Використовуючи цю інформацію, алгоритм може регулювати ваги кожного з'єднання, щоб значно зменшити значення функції помилки. У цьому дослідженні функція `mlp` відповідає нейромережам MLP для прогнозування часових рядів, що виконуються за допомогою пакета `nnfor`.

Мережі довгострокової короткочасної пам'яті (LSTM) - це особливий тип нейромереж з періодичним глибоким навчанням, які вивчають залежність порядку в задачах прогнозування послідовності. LSTM був запроваджений Хохрейтером та Шмідхубером у 1997 р., і зараз він широко використовується в різних дослідженнях та проектах. Типовий RNN використовує послідовну інформацію. Ці мережі описуються як періодичні, оскільки вони використовують свій внутрішній стан для обробки послідовностей змінної довжини входів. Стандартному RNN важко переносити інформацію з попередніх етапів часу на наступні, якщо послідовність занадто довга, оскільки це може виключити важливу інформацію з самого початку. Тому LSTM має перевагу в тому, що інформація може запам'ятовуватися протягом тривалого періоду часу. На відміну від традиційних FNN, LSTM має з'єднання з зворотним зв'язком, завдяки чому вихід з попереднього кроку подається як вхід на поточному кроці. Звичайний блок LSTM включає в себе комірку, вхідний затвор, вихідний затвор і затвор забуття. Клітина запам'ятовує значення протягом довільного інтервалу часу, і три ворота регулюють потік інформації в клітинку та з неї. У цьому дослідженні було використано пакет `keras` R, щоб згадати TensorFlow для проведення аналізу LSTM. TensorFlow був

розроблений командою Google Brain і випущений у 2015 році. Це безкоштовна бібліотека програмного забезпечення з відкритим кодом для методів машинного навчання, особливо глибоких нейронних мереж.

Карти тепла можна створювати для відображення змін у політичних заходах щодо пандемії COVID-19 у часі. Градієнтні кольорові смуги відображають зміни у заходах на різних рівнях та підтримку, отриману у вигляді фінансової допомоги та інвестицій. Графік роботи, представлений по горизонтальній осі, буде оновлюватися щодня. Накопичувальні та щомісячні записи представлені відповідно до гістограм та лінійних діаграм. Ця система також надає можливість завантаження для зацікавлених країн та подібні послуги з динамічним рейтингом загальної кількості підтверджених випадків та смертей та тенденціями до зниження пандемії COVID-19. Наступна проста формула регресії використовується для вивчення тенденцій спадання з динамічними інтервалами часу:

$$y_i = \alpha + \beta x_i$$

де  $\beta$  - нахил, що представляє тенденцію збільшення або зменшення.

У цьому дослідженні CPAIS було розроблено для вивчення варіацій, тенденцій та прогнозів, пов'язаних з пандемією COVID-19 у кількох країнах. Доступний розкривний список для вибору країни. Структура CPAIS - від збору та попередньої обробки даних до застосування моделі глибокого навчання, прогнозування та візуалізації даних — представлена на Рисунку 1. Вона включає комбінацію двох наборів даних, побудову баз даних для прогнозування глибокого навчання та статистичного аналізу, чотири статистичні моделі або моделі глибокого навчання для прогнозування та функції інтерфейсу для візуалізації даних.

## **2.2 Моделі глибокого навчання та статистичного навчання**

Кількість підтверджених щомісячних випадків у Сполучених Штатах досягла локального піку в липні 2020 року та ще одного піку - 6 368 591 у грудні 2020 року. Щодо США, кількість відновлених випадків після 14 грудня 2020 р. Не реєструється в базі даних сховища даних COVID-19. Загальна чисельність населення для кожної з 15 країн у 2020 році також вказана в таблиці. Динамічна теплова карта з політичними заходами наведена на Рисунку 2.2, де зображено зміни у заходах COVID-19 для кожної країни, на прикладі Австралії. Загалом 19 заходів було включено до трьох основних розділів політики (тобто політики стримування та закриття, економічна політика та політика системи охорони

здоров'я). Економічна політика має найменшу кількість заходів, і лише 4 із 19 заходів є постійними заходами, пов'язаними з фінансовою підтримкою чи інвестиціями.

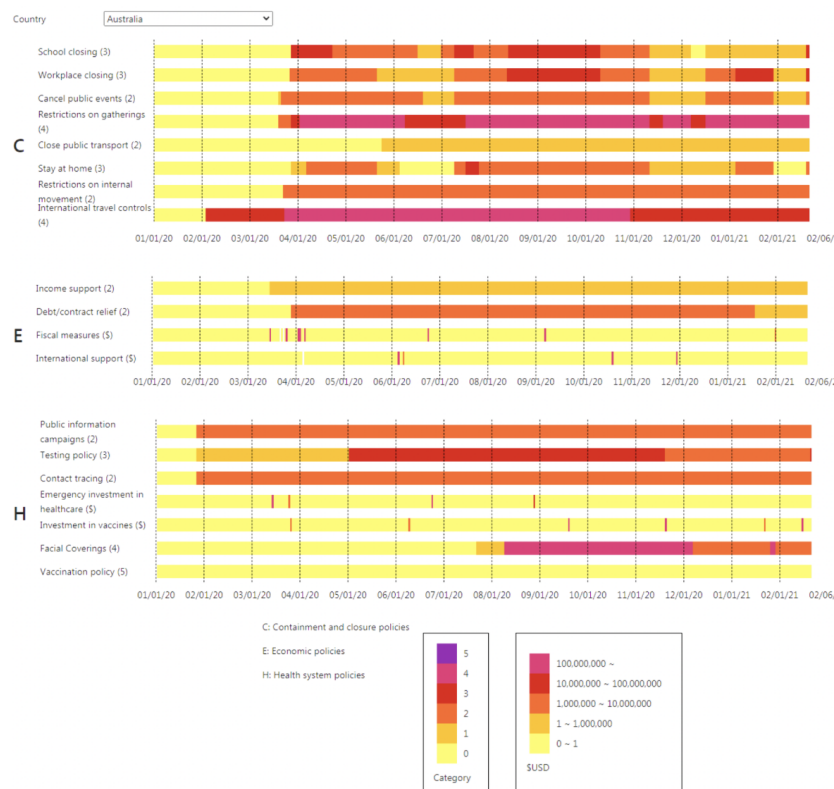


Рисунок 2.2 - Інтерфейс динамічної теплової карти з політичними заходами на веб-сайті COVID-19 Pandemic AI System (CPAIS).

Для спрогнозування COVID-19 використовувались моделі глибокого навчання та статистичного навчання. Функція полегшує 14-денне прогнозування за допомогою чотирьох потужних алгоритмів, можна подивитись на Рисунок 2.3. ARIMA - це статистична модель навчання з регресією часових рядів; інші моделі - це алгоритми глибокого навчання нейронних мереж з одним прихованим шаром, декількома прихованими шарами або повторюваними методами. Невелике значення похибки вказує на ідеальну відповідність даним, але порівняння між різними країнами не мало сенсу, оскільки вони мали різні базові лінії на основі їх популяції. Для більшості країн LSTM продемонстрував кращу точність прогнозу з меншою кількістю помилок, ніж інші моделі. Ефективність ARIMA, FNN та нейронної мережі MLP не були стабільними, оскільки їх точність прогнозу була конкурентоспроможною лише LSTM для деяких конкретних країн. Наприклад, LSTM продемонстрував найкращу точність прогнозу для Канади. RMSE, MAE та MAPE становили 2272,551, 1501,248 та 0,2723075, відповідно. ARIMA (RMSE = 317,53169; MAPE = 0,4641688) та FNN (RMSE = 181,29894; MAPE = 0,2708482) продемонстрували кращі показники для Південної Кореї.

На рисунку 2.3 представлена описова статистика для конкретних країн. На веб-сайті можна одночасно порівняти три країни та налаштувати період. Користувачі можуть вибрати країни, які їх цікавлять, і порівняти дані, пов'язані з COVID-19. Для кожної відповідної країни формується лінійна діаграма, що показує кількість підтверджених випадків, одужання та смертності за місяць. Крім того, на веб-сайті також пропонується глобальне порівняння.

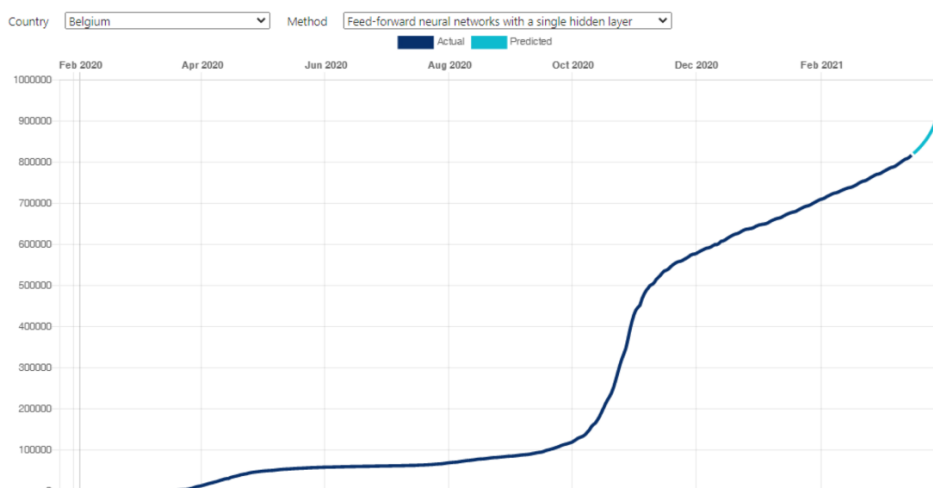


Рисунок 2.3 - Інтерфейс системи пандемічного ШІ COVID-19 (CPAIS) для моделей прогнозування машинного навчання, що полегшує 14-денне прогнозування COVID-19. Сюжет показує криву для моделювання глибокого навчання загальних сукупних підтверджених випадків.



Рисунок 2.4 - Інтерфейс описової статистики для вибраних країн із налаштуванням на веб-сайті системи інтелектуальної індукції пандемії COVID-19 (CPAIS). CSV: значення, розділені комами.



Користувачі можуть класифікувати 171 країну на основі п'яти різних параметрів:

- Кількість підтверджених випадків;
- Підтверджених випадків за відсотком населення;
- Кількість підтверджених смертей;
- Підтверджених смертей за відсотком населення;
- Тенденція до зниження.

На Рисунку 2.5 наведено приклад того, як 20 країн, що входять до списку, можна класифікувати за підтвердженими випадками за відсотком населення. Щодо налаштування, функція ранжирування є гнучкою. Користувач може змінити вибрані країни та певний період часу.

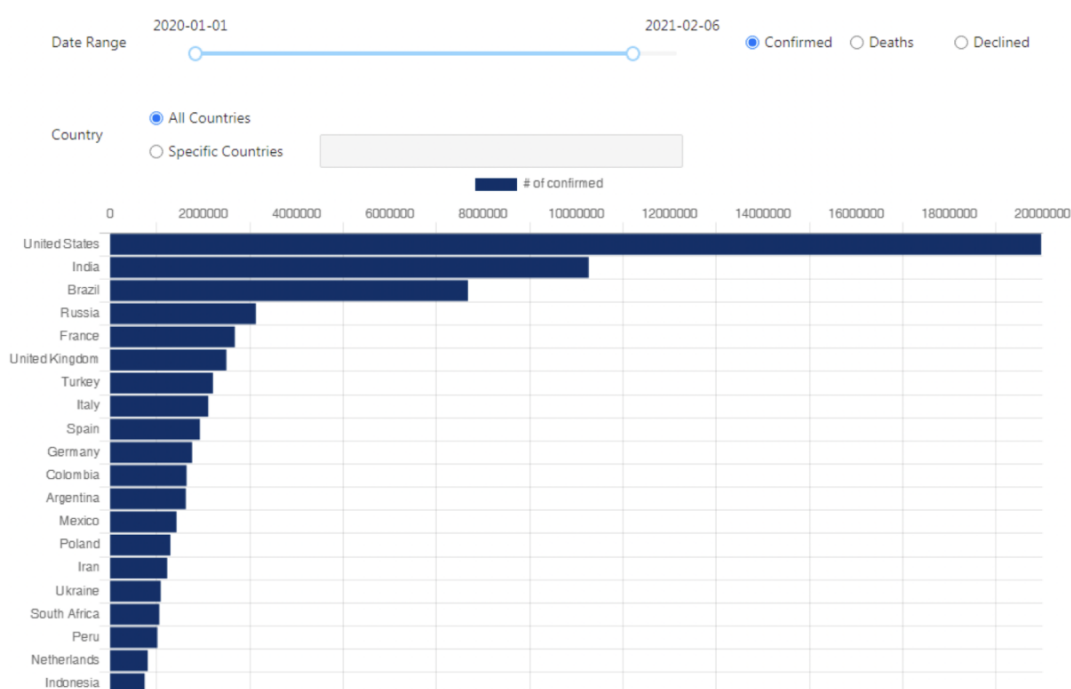


Рисунок 2.5 - Інтерфейс для ранжирування вибраних країн з налаштуванням на веб-сайті системи пандемічного ШІ COVID-19 (SPAIS).

Поєднання даних про захворюваність на COVID-19 та політичні заходи можуть бути використані для вивчення взаємозв'язку між прогресуванням пандемії COVID-19 та урядовими зусиллями з профілактики епідемій. SPAIS може допомогти користувачам визначити, чи успішні заходи політики запобігають передачі COVID-19. Згідно з доповіддю, опублікованою Інститутом міжнародної політики Лоуї, рейтингове порівняння результатів діяльності країн з управління пандемією COVID-19 показує, що Нова Зеландія, В'єтнам і Тайвань входять до трійки країн з найвищими середніми показниками за їхніми шістьма показниками. Крім того, Нова Зеландія та Тайвань успішно контролювали спалах COVID-19 без міжнародної фінансової підтримки (рисунки S1-S3 у Мультимедіа Додаток 1).

Зокрема, Нова Зеландія негайно запровадила політику боротьби з інфекцією та закриття з гнучкою адаптацією заходів; крім того, Тайвань дотримувався суворих вказівок щодо міжнародних поїздок, які не лише сприяли боротьбі з інфекцією, а й робили непотрібними жорсткі заходи, описані в політиці стримування та закриття. Крім того, обидві країни доклали великих зусиль для максимального впровадження політики тестування та відстеження контактів протягом 2020 року. У цьому зв'язку обидві країни є видатними прикладами. Яскраві теплові карти в CRAIS ілюструють коливання показників, що залежать від часу, і допомагають користувачам відстежувати зміни та наслідки політичних заходів у кожній країні.

Для цілей прогнозування використовувались декілька методів навчання AI з часовими рядами. Як статистичне навчання, так і моделі глибокого навчання продемонстрували ефективну ефективність для різних країн. Хоча значення не є абсолютними, вони порівнянні між країнами з різною загальною чисельністю населення. Порівняно з результатами минулого дослідження, показники для тієї ж моделі та країни були кращими в цьому дослідженні, оскільки в нашу систему були включені більш широкі дані часових рядів. Крім того, 14-денне прогнозування тенденцій COVID-19 може подавати корисне попередження, яке допоможе урядам та експертам зменшити рівень захворюваності на COVID-19. Крім того, різні методи навчання ШІ мають унікальні переваги.

Відповідно до теореми про розкладання Вольда, моделі авторегресійної ковзної середньої теоретично достатньо для опису регулярного стаціонарного часового ряду. Можна нестационарний часовий ряд змінити на стаціонарний, наприклад, використовуючи диференціювання. Як зазначалося раніше, моделі ARIMA мають три компоненти: авторегресію, інтеграцію та ковзну середню. Вони застосовуються до даних із свідченнями нестационарності середнього значення, завдяки чому початковий етап диференціювання може застосовуватися один або кілька разів для усунення нестационарності середньої функції в тренді. Було використано функцію `auto.arima` для R, щоб вибрати найкращу модель відповідно до інформаційного критерію Акакайке, виправленого інформаційного критерію Акакайке або значення інформаційного критерію Байеса; функція `auto.arima` також здійснює пошук моделі в межах передбачених обмежень порядку. FNN подібний до ARIMA, оскільки встановлена модель є аналогом моделі авторегресії (p), де p - порядок, але з нелінійними функціями для несезонних даних у цьому дослідженні. Тому її позначають як модель авторегресії нейронної мережі (p, k), яка називається NNAR, де k представляє кількість прихованих вузлів. Ось чому для деяких країн ARIMA та FNN дали подібні результати для точності прогнозу. Відмінності між

двома моделями все ще існують; помилку можна зменшити лише для FNN, збільшивши кількість ітерацій, але в результаті час ітерації буде збільшено.

Можливості нейронних мереж пояснюються ієрархічною або багатошаровою структурою мереж. Структура даних може включати функції в різних масштабах або роздільній здатності та об'єднувати їх у функції вищого порядку. Після повторення процесу навчання протягом достатньої кількості навчальних циклів мережа перейде в певний стан, де термін помилки досить малий. Узагальнення та толерантність - дві основні характеристики. По-перше, нейронні мережі дозволяють узагальнення, оскільки вони можуть класифікувати як невідомі, так і відомі закономірності з однаковими відмітними ознаками. По-друге, нейронні мережі високо стійкі до несправностей. Через розподілену природу вони продовжуватимуть функціонувати, навіть якщо значна частина нейронів та взаємозв'язків вийде з ладу. Взагалі, збільшення кількості прихованих вузлів може підвищити ефективність прогнозування, а збільшення кількості мереж для навчання може призвести до прогнозування ансамблю.

Основна ідея LSTM полягає у стані комісії - горизонтальній лінії, яка проходить по ланцюжку з інформацією, що протікає порядком, можна побачити на Рисунку 2.6. Крім того, LSTM може видалити або додати інформацію до стану комісії, керованих воріт, які через них можуть передавати інформацію. Вони складаються з сигмоподібного нейронного шару та операцій точного множення. Мережі LSTM є потужними для оперативного прогнозування серії даних, які можуть бути між поділами в часових рядах можуть бути затримки невідомої тривалості. Отримавши, у порівнянні з іншими традиційними RNN у цьому дослідженні, мережа LSTM не має проблем зі зведенням градієнта. Таким чином, LSTM отримує переваги щодо відновної невміння до інтервалу часу та зменшення кількості помилок при прогнозуванні порівняно з іншими методами.

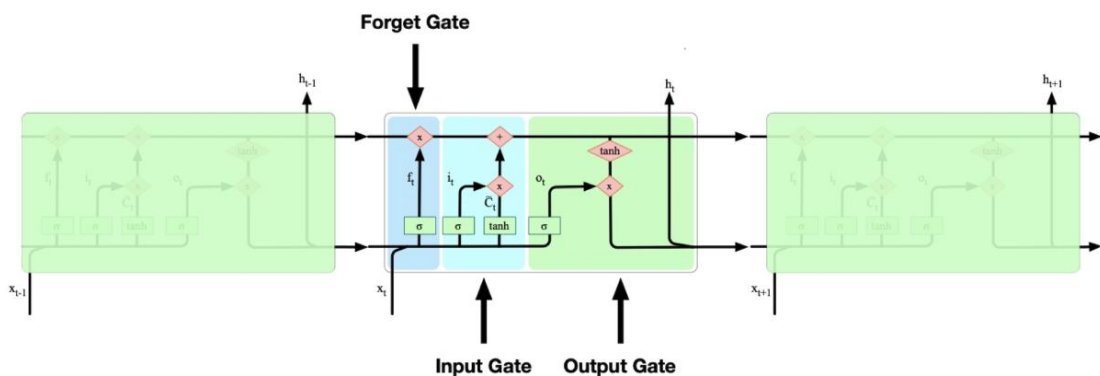


Рисунок 2.6 - Схема нейронної мережі довготривалої пам'яті з трьома функціональними ворітами.

У CPAIS включені довгострокові кумулятивні записи підтверджених випадків, необхідність та сміливість. Крім того, щомісячно подаються показники цих показників для кожного місяця. Таким чином, короткострокові тенденції можна дослідити за допомогою цієї системи. Користувачі можуть порівнювати три або більше країн і візуалізувати відновлену частоту побоювань на COVID-19 протягом певного періоду часу. Короткострокові та довгострокові тенденції можна одночасно розглядати. У попередніх дослідженнях для прогнозування було включено обмежену кількість країн. Наша система містить 171 країну та надає інформацію про політичні заходи. Крім того, можливі візуалізація даних, статистичне та глибоке навчання для прогнозування захворюваності та індивідуального ранжування. На основі своїх цілей користувачі можуть вибрати назви країн і періоди часу. Схожі культурні традиції, сусідні географічні характеристики та високочастотна торгівля також можуть служити приватними рисами. Тому, наша система розкриває відомий рейтинг для вивчення ефективності стратегії управління COVID-19, реалізованої в 2020 році. Таким чином, CPAIS - це комплексна послуга на базі ШІ, яка доступна в Інтернеті. Він спирається на великі дані та пропонує візуалізацію даних, прогнозування на основі глибокого навчання та індивідуального порівняння. Ця система може бути використана для дослідження тенденцій прогресування COVID-19.

Наскільки нам відомо, це перша веб-система машинного навчання, яка може досліджувати варіації, тенденції та прогнози, пов'язані з пандемією COVID-19, у 171 країні. Ця пілотна система все ще має кілька обмежень. По-перше, ця база даних значною мірою покладається на вихідні бази даних і має подібні обмеження з вихідними базами даних. Наприклад, у вихідних базах даних не враховувалася кількість пацієнтів з COVID-19, які подорожували за кордон, і це може призвести до неточного аналізу для невеликої кількості країн. Однак вважається, що кількість пацієнтів із COVID-19, які подорожували за кордон, невелика, оскільки більшість країн наклали негативні тести на COVID-19 або докази вакцинації перед тим, як дозволити мандрівнику в країну. По-друге, CPAIS не можна оновлювати щодня, якщо вихідні бази даних не оновлюються. Наприклад, наразі кількість вилучень у Сполучених Штатах востаннє оновлено 14 грудня 2020 р. Тож кількість вилучень у Сполучених Штатах може бути неточною. Нарешті, оскільки основною метою цієї платформи є консолідація необроблених даних, отриманих з різних баз даних, та пов'язаних із ними заходів щодо реалізації пандемічної політики, краще використовувати для видобутку тексту, місцевих звітів та інформації, отриманої з медичної системи певної країни, для подальшої оцінки.

Загалом CRAIS збирає та узагальнює інформацію про пандемію COVID-19 та пропонує візуалізацію даних та прогнозування на основі глибокого навчання. Це може бути корисним та послідовним довідковим ресурсом для будь-яких серйозних спалахів або епідемій, які можуть виникнути в майбутньому. Крім того, інформація про вакцину також зберігається в нашій системі. Він може бути використаний для оцінки ефективності вакцини в різних країнах у майбутньому. Більше того, прогнози машинного навчання на 2 тижні можуть служити попереджувальними знаками та висвітлювати сучасні тенденції епідемії, які виявилися за допомогою методів ШІ. Можна додати, що CRAIS можна використовувати для узагальнення кількох факторів, які можуть вплинути на ефективність профілактики епідемій та передбачити наступний серйозний спалах.

### **2.3 Аналіз впливу COVID-19 на всесвітню економіку**

Пандемія covid-19 вимагає втручання, що зменшують фізичний контакт між людьми, що негативно впливає на нашу економіку. За деякими оцінками, чверть усіх робочих місць в економіці потребує фізичної взаємодії, і, отже, безпосередньо впливає пандемія. Це добре помітно в медичній галузі, де працівники та пацієнти часто стикаються між собою і ризикують передати хворобу. У кількох країнах медичні працівники пережили одні з найвищих випадків захворювання на covid-19. Більше того, оскільки пацієнтам рекомендували відкладати необов'язкові візити та процедури, медичні працівники багатьох країн також зазнали колосальних втрат доходу.

Економічною мовою, covid-19 додав "тіньову вартість" на робочу силу, яка вимагає близькості. Ця тіньова вартість відображає доларовий еквівалент усіх витрат, пов'язаних із підвищеним ризиком передачі хвороби, включаючи витрати на адаптацію, необхідну для covid-19. Він складається із втрат як днів життя, скоригованих на якість, від підвищеної захворюваності, так і років життя, скоригованих на якість, від збільшення смертності, а також витрат на заходи для зменшення цих ризиків, таких як додаткове захисне обладнання та заходи щодо дистанції для працівників. Деякі сектори нестимуть збільшені витрати через зміну фізичних механізмів, в яких відбувається виробнича та інша взаємодія, так що може бути соціальне дистанціювання. Звичайно, зрозуміло, ці заходи вживаються для зменшення розповсюдження хвороби: за деякими оцінками, соціальна вартість одного додаткового випадку covid-19 протягом пандемії становить 56 000 доларів (40 000 фунтів стерлінгів; 46 000) до 111 000 дол.

Ці тіньові витрати на робочу силу також прискорюють розробку та впровадження нових технологій для автоматизації людської праці. Одним із прикладів є все більше використання телемедицини. В даний час телемедицина надається таким чином, що змінює формат надання допомоги, але залишає роль лікарів майже незмінною. Однак це зменшує потребу у працівниках, які надають допоміжні послуги і які, як правило, мають нижчу заробітну плату, ніж лікарі - наприклад, фронт-офіс чи прибиральний персонал - тим самим збільшуючи нерівність. Більше того, надалі це може також зробити можливим надання медичних послуг з інших країн, що до цього часу було важким, і, отже, зменшити попит на лікарів у країнах з високим рівнем доходу.

Додаткові інвестиції, наприклад, підключені до Інтернету пристрої, такі як термометри, імпульсні оксиметри пальців, манжети для артеріального тиску, цифрові стетоскопи та електрокардіографічні прилади, можуть ще більше змінити надання медичної допомоги, а також можуть зменшити попит на медсестер.<sup>45</sup> Такі технології вже зробили можливим створити «віртуальні палати» для пацієнтів з covid-19.<sup>6</sup> Але навіть після того, як covid-19 контролюється, медичні працівники враховуватимуть ризик майбутніх пандемій, вибираючи, в які технології інвестувати. AI), такий як чат-бот Babylon Health, передбачає можливе майбутнє, коли медичні функції, які традиційно виконують лікарі, також можуть бути автоматизовані. Це зменшило б попит на робочу силу та створило б цілий новий набір потенційних проблем.

У минулому такі ризики кібербезпеки, як комп'ютерні віруси, стримували автоматизацію, особливо в медичному секторі, де конфіденційність та безпека становлять особливу стурбованість. Іронічно, що людський вірус зараз вирівнює умови і примушує автоматизуватися, оскільки він зменшив апетит до працевлаштування людей. Ці розробки можуть призвести до зменшення попиту на робочу силу та заробітної плати в економіці, в тому числі в галузі охорони здоров'я. Однак скорочення робочої сили не є неминучим. Технологічним прогресом у ШІ та суміжних областях можна керувати таким чином, щоб переваги технологічного прогресу були широко розподілені.

Страх втрати робочих місць супроводжував технічний прогрес з часів промислової революції. Історія прогресу була однією з невинних хвиль на ринку праці, завдяки яким прогрес робив старі робочі місця надлишковими та створював нові. Це хвилювання завжди було боляче для переміщених працівників, але раніше економісти вважали, що за нові робочі місця, створені внаслідок прогресу, буде вища оплата праці, ніж ті, що стали зайвими, щоб прогрес зробив працівників вигіднішими, коли вони пройдуть коригування.

Найбільш корисний спосіб проаналізувати вплив нової технології на ринки праці - не розглядати, чи знищує вона робочі місця в короткостроковій перспективі - багато технологій це зробили, хоча в довгостроковій перспективі вони виявились корисними для працівників. Натомість найбільш корисно класифікувати наслідки технічного прогресу залежно від того, чи використовується вони робоча сила, чи економія праці, тобто збільшують чи зменшують загальний попит на робочу силу за певних заробітних плат і цін. Наприклад, автоматизація багатьох процесів, пов'язаних з медичними консультаціями, як у прикладі телемедицини, швидше за все, заощаджує робочу силу, тоді як нові медичні методи для поліпшення стану здоров'я пацієнтів, швидше за все, будуть робочою силою, якщо їх виконує людина. у довгостроковій перспективі, коли ринки пристосовуються, зміни в попиті на робочу силу в основному відображаються на заробітній платі, а не на кількості створених або втрачених робочих місць.

Загалом, технічний прогрес після Промислової революції використовував робочу силу - він стрімко збільшив попит на робочу силу, що призвело до масового зростання середньої заробітної плати та матеріального багатства в передових країнах. Причиною було те, що інновації збільшили продуктивність праці робітників, зробивши їх здатними виробляти більше за годину, а не замінюючи робочу силу роботами.

Однак нещодавно економічна картина була менш доброякісною: значна частина робітників у США - наприклад, виробничі та неконтролюючі працівники - отримують зараз нижчу заробітну плату (з урахуванням інфляції), ніж у 1970-х. Більше того, хоча незрозуміло, чи відповідає ця висновок у решті світу, частка економічного виробництва в США, що надходить працівникам, а не власникам капіталу, за останні півстоліття зменшилася з 65% до менш ніж 60% . Найбільше постраждали нижчі кваліфіковані працівники. Багато останніх технологій автоматизації витіснили працівників із роботи таким чином, що зменшили загальний попит на людську працю.

Досягнення ШІ можуть сприяти зростанню загального процвітання, але також існує ризик того, що вони прискорять тенденцію останніх чотирьох десятиліть. Визначальним атрибутом штучного інтелекту є автоматизація останньої сфери, в якій працівники людини мали порівняльну перевагу над машинами - наше мислення та навчання. І якщо пандемія covid-19 додасть додаткові стимули для інновацій, що заощаджують працю, економічні наслідки будуть ще болючішими ніж у минулих епізодах технічного прогресу. Коли економіка розростається, а прогрес упереджується проти праці, працівники все ще

можуть відчувати помірне збільшення своїх доходів, хоча відносна частка виробленої ними продукції зменшується. Однак у той час, коли економічне виробництво по всьому світу падає через наслідки covid-19, зменшення відносної частки продукції, заробленої працівниками, означає, що їх доходи падають швидше, ніж решта економіки. А некваліфіковані працівники фізичної роботи, які перебувають на нижчих щаблях розподілу заробітку, ймовірно, будуть найбільш сильно постраждали.

Додатковим аспектом цифрових технологій, таких як ІІ, є те, що вони породжують те, що часто називають явищем "суперзірки", що може призвести до подальшого збільшення нерівності. Цифрові технології можна застосовувати майже за незначну ціну, як тільки вони будуть розроблені. Тому вони породжують природні монополії, що призводить до домінуючих ринкових позицій, за допомогою яких фірми суперзірок обслуговують значну частину ринку - або тому, що вони кращі за будь-яких конкурентів, або тому, що ніхто навіть не намагається дублювати свої зусилля та конкурувати. Ці ефекти суперзірки добре відомі в індустрії розваг. Наприклад, у музичній індустрії суперзірки мають сотні мільйонів шанувальників і пожинають пропорційну винагороду, але доходи музикантів у списку швидко падають. Більшість нагород впливає на вершину. І емпіричні робочі документи, що ці ефекти суперзірок зіграли важливу роль у зростанні нерівності в останні десятиліття.

Подібний механізм може незабаром застосувати в медицині, прискорений пандемією covid-19. Типовим прикладом є рентгенологія. Якщо одна з провідних світових компаній, що займається візуалізацією, розробить систему штучного інтелекту, яка зможе читати та надійно інтерпретувати мамографію краще, ніж люди, вона стане «суперзіркою» у цьому секторі та замінить завдання зчитування мамографій для тисяч рентгенологів. Оскільки вартість обробки додаткового набору зображень близька до нуля, будь-який прибуток після окупності первинних інвестицій в систему призведе до високої норми прибутку, і компанія, ймовірно, отримає значні економічні вигоди, принаймні до тих пір, поки її інтелектуальна власність захищена патентами чи комерційною таємницею. (Структура режиму інтелектуальної власності є важливим фактором, що визначає ступінь нерівності, спричиненої обговорюваними тут економічними перетвореннями.) Чим ширше будуть розповсюджуватися такі засоби діагностики та прийняття рішень, тим більше медичний сектор перетвориться на галузь суперзірок.

Економічні сили продовжують швидко розвивати ІІ, а Covid-19 додає цим силам попутний вітер. Зараз завдання полягає у формуванні форм, які будуть мати ці досягнення, щоб забезпечити бажаний їх вплив як на пацієнтів, так і на



медичних працівників. Ставки високі, оскільки вибір, який ми робимо зараз, матиме довготривалі наслідки.

Ми добре розуміємо, що відбувається в одній крайності: якщо напрямок прогресу визначатиметься суто ринковими силами, не враховуючи загального добробуту людини, наше технологічне майбутнє буде формуватися через недоліки та провали ринку. Ринки можуть забезпечити силу для підвищення ефективності, але не звертають уваги на проблеми розподілу, такі як шкідливі наслідки прогресу в економії праці або явище суперзірки. Відповідальні особи, які приймають рішення, повинні застосовувати технології, що підтримують активну роль людини та зберігають роль медичних працівників усіх освітніх рівнів. Наприклад, медичні системи штучного інтелекту можуть бути розроблені як орієнтовані на людину інструменти, що забезпечують підтримку прийняття рішень, або вони можуть бути розроблені для автоматизації людських завдань. Вони також повинні зосередитись на наданні високоякісної допомоги та цінності пацієнтам з обмеженими фінансовими можливостями, а не просто на обслуговуванні пацієнтів відповідно до їх платоспроможності.

Невдачі на ринку поширені як в інноваціях, так і в охороні здоров'я, і тим більше на перетині цих двох. Ринки заохочують додаткові аванси, які можуть не надати великої цінності суспільству. Вони недостатньо стимулюють масштабні прориви, які є найбільш соціально вигідними. І як показала пандемія covid-19, вони недооцінюють переваги профілактичних дій, включаючи профілактичні дії проти невеликої ймовірності, але екзистенціальних ризиків.

Невдачі на ринку іноді посилюються державною політикою, яка збільшує вартість робочої сили відносно капіталу, ставлячи людей у не вигідне становище щодо машин. Прикладами можуть бути низькі податки на капітал (особливо прибуток від капіталу) щодо праці та штучно низькі процентні ставки, що панували після фінансової кризи 2008 року (хоча низькі процентні ставки також стимулюють сукупний попит, що вигідно для робітників).

Наші установи та норми важливі для взаємодії з ринковими стимулами для технічного прогресу. Найвидніше, що наша система прав інтелектуальної власності, надаючи тимчасову монопольну владу винахідникам, має сприяти інноваціям. Але часто це має протилежний ефект - перешкоджає доступу до існуючих знань та ускладнює вироблення нових ідей. Більше того, за рахунок стримування конкуренції зменшуються як інновації, так і доступ до переваг, що відбуваються. Це аргументи для обмеження обсягу та тривалості прав інтелектуальної власності. Нарешті, ринки за своєю суттю погано доставляють людський елемент, який так важливий у медичній допомозі. Ринки не

винагороджують належним чином співчуття та співчуття, які медичні працівники надають своїм пацієнтам, і насправді стимулюють їх обмежувати. Якщо наші технологічні рішення визначатимуться виключно ринком, вони відображатимуть однакові упередження, і, швидше за все, це вплине на догляд за пацієнтами. Дуже важливо, щоб ті, хто приймає рішення, діяли так, щоб наші технологічні рішення відображали наші людські цінності.

В недавньому дослідженні MarketsandMarkets прогнозують, що до 2025 року світовий ринок ШІ досягне 200 мільярдів доларів. До 2030 року прогнози PWC стверджують, що технології ШІ матимуть вплив на світовий ВВП на 15 трильйонів доларів. Коротше кажучи, вплив ШІ протягом наступного десятиліття буде монументальним.

Основними рушійними силами зростання ринку штучного інтелекту є: великі обсяги даних, прийняття хмарних додатків та збільшення попиту на розумних віртуальних помічників, таких як чат-боти.

Очікується, що ринок Азіатсько-Тихоокеанського регіону (APAC) стане першим регіоном, де спостерігається найвище зростання технологій та додатків ШІ. У цій частині світу все більш широке впровадження технологій поглибленого навчання, автоматизованого роботизованого процесу (RPA) та обробки природних мов (NLP) буде найбільш помітним у галузях законодавства, сільського господарства, роздрібною торгівлі, маркетингу, бухгалтерії та фінансів.

В умовах поточної кризи, спричиненої COVID-19, інновації, керовані людськими діями, стануть ключовими у використанні потенціалу застосувань ШІ у різних галузях. З огляду на це, ШІ продовжує залишатися пріоритетом для керівників підприємств у всьому світі, які прагнуть сприяти організаційному зростанню в умовах безпрецедентної кількості невизначеності.

Хоча інструменти ШІ здатні допомогти людям (і бізнесу) вирішити нагальні проблеми, порушені пандемією, сама технологія не може творити чудес. Креативні професіонали, що володіють знаннями та досвідом технології, є ключем до використання справжнього потенціалу ШІ. Для боротьби з новими проблемами, які представляє вірус, вирішення проблеми починається зі збору нових навчальних даних у сучасних умовах. Для людей, а також систем, що працюють на основі штучного інтелекту, будь-яка нова інформація про поточну глобальну ситуацію виявиться цінною для інформування розумних рішень у майбутньому. Хоча ефективний обмін інформацією є ключовим, цифрові інновації за допомогою ШІ повинні бути етичними, екологічно обґрунтованими та стійкими для бізнесу майбутнього. Щоб сприяти економічному відновленню, технології ШІ повинні використовуватися для вирішення основних проблем, а не загострювати їх.

Ми не можемо повернутися до „нормальної” економіки після впливу COVID-19. Це означає, що основне питання, з яким зараз стикаються керівники підприємств та керівники ключових рішень: що робити далі?

Як ми вже зазначали, ШІ та інші інтелектуальні цифрові технології мають величезну потужність та потенціал. За даними Світового економічного форуму: "Кожна унція технологічних інновацій та винахідливості, задіяних для боротьби з цією пандемією, наближає нас на крок до її подолання".

Але як ШІ на сьогоднішній день пандемії був ключовим для міжнародного економічного зростання?

ШІ має потенціал для покращення результатів діяльності компанії, але він також може прискорити нерівність та збільшити існуючі робочі ролі. Хоча деякі сектори, безсумнівно, побачать скорочення робочих місць внаслідок ШІ та автоматизації, підприємства, які активно допомагають своїм працівникам підвищувати кваліфікацію, будуть важливою частиною вирішення будь-якої нестабільності чи втрати роботи, спричиненої ШІ.

На макроекономічному рівні технологія ШІ забезпечує більший потенціал для прогнозування майбутніх результатів - і, як результат, більшу визначеність. Протягом останніх кількох місяців три ключові сфери впливу, які ШІ справив на глобальне економічне зростання, включають:

1. Швидка наука про дані збільшила швидкість прийняття рішень для національних урядів.
2. Багатоагентне моделювання та планування сценаріїв сприяють більшій впевненості щодо майбутнього за межами COVID-19 для керівників.
3. Відсутність історичних даних призвело до зростання модельного ШІ, що покращує неадекватність глобальних систем, процесів, управління та поведінки.

Ще до COVID-19 ШІ вже був визнаний головною силою у розробці та управлінні глобальними ланцюгами створення вартості. Подібним чином технологія була розгорнута для вирішення проблем, пов'язаних з вірусом, ШІ застосовується для перегляду майбутніх тенденцій, таких як зміни споживчого попиту, а також для ефективного управління ризиками на кожному етапі у всьому ланцюжку поставок.

Завдяки можливості бізнесу краще управляти складними та різними виробничими процесами, ШІ сприяє підвищенню ефективності та загальної прибутковості. Наприклад, для керівників складських приміщень ШІ може допомогти передбачити попит на продукцію, одночасно покращуючи точність виготовлення та доставки вчасно.

Спалах Covid-19 - це криза охорони здоров'я, яка має все більший вплив на світову економіку. Пандемія спричинила спад, рекордний рівень безробіття та безпрецедентний рівень боргу у всьому світі.

Лише за даними Організації Об'єднаних Націй лише в 2020 і 2021 роках виплати країнам, що розвиваються, лише за їх державний зовнішній борг становитимуть від 2,6 до 3,4 трильйонів доларів. Світовий банк прогнозував, що коронавірус штовхає від 40 до 60 мільйонів людей у крайню бідність. На сьогодні заклики до міжнародної солідарності мало відчутної підтримки, і зростає ризик дефолту заборгованості країни та банкрутства підприємств, а також пов'язані з цим людські витрати.

Країни потребують нових моделей економічного зростання, і ШІ (штучний інтелект), що працює на базі 5G, дедалі більше стає на першому плані, оскільки країнам, компаніям та приватним особам потрібно серйозно подумати та підготувати наступний пост Covid-19. Рецесія коронавірусу сприяє більшому впровадженню нових бізнес-моделей, які використовує ШІ.

Всі найцінніші компанії у світі використовують обчислювальні соціальні науки, і понад 80% щоденних ходів на фондовому ринку США, як вважають, сьогодні є машинною технологією штучного інтелектуальної торгівлі.

Початковий розвиток ШІ вже суттєво вплинув на кілька нішових областей, але зараз він, схоже, впливає майже на всі частини економіки та, дійсно, на всі аспекти нашого життя.

Прикрою правдою глобальної пандемії коронавірусу та еволюції ШІ є те, що вона найбільше впливає на найуразливіших людей, низькокваліфікованих людей у суспільстві та найбідніші країни.

Пандемія вже спорожнює кол-центри, оскільки чат-боти ШІ збільшуються, щоб замінити працівників колл-центрів. Хоча кол-центри вже давно стали межею автоматизації робочих місць, пандемія прискорила процес.

Оскільки близько 38 мільйонів американців втратили роботу через Covid-19, і деякі з них можуть не повернутися, американські компанії, які перебувають на передовій ШІ, лише стають все більш потужними.

На світовому ринку акцій лідирують американські акції, а в межах цієї великої межі американські технології та так звані акції FAANG. П'ять найцінніших компаній у світі все частіше становлять усі компанії зі штучним інтелектом. Facebook, Apple, Amazon, Netflix та Google (алфавіт).

У Китаї така ж ситуація - Baidu, Tencent та Alibaba - це компанії зі штучним інтелектом. З точки зору ринкової капіталізації Alibaba та Tencent є найціннішими

компаніями, що розміщуються в Китаї, і обидві є передовими компаніями зі штучним інтелектом.

У 2017 році китайський уряд оголосив про плани "очолити світ" у галузі штучного інтелекту до 2030 року. Це повідомлення викликало значне занепокоєння в США та інших регіонах щодо масштабів прагнень Китаю та міри, в якій комуністична партія може використовувати ШІ для посилення жорсткості контроль за своїми громадянами та розвиток більш досконалих військових можливостей.

Через політичний тиск Китайської Народної Республіки Тайвань не є членом ВООЗ (Всесвітньої організації охорони здоров'я), схоже, на диво добре впорався з пандемією. На момент написання статті Тайвань мав 442 випадки, 7 смертей та 420 одужалих від коронавірусу, незважаючи на те, що уникав блокування.

Тайвань також є енергійною демократією, яка працює у тісній співпраці з фірмами приватного сектору, стаючи головним напрямком штучного інтелекту. Міністерство науки і технологій Тайваню (MOST) має намір побудувати інноваційну екосистему зі штучним інтелектом, що включає різні аспекти ШІ, такі як таланти, технології, сфери діяльності та галузі, не тільки спрямовані на досягнення розумного суспільства, але і роблять Тайвань ключовим гравцем у глобальному ланцюжку створення вартості ШІ.

Зараз Тайвань посідає третє найкраще місце у світі для інвестування (лише Швейцарія та Норвегія) і найкраще місце в Азії перед Сінгапуром та Південною Кореєю, згідно з трирічним звітом американського бізнес-середовища Risk Intelligence SA (BERI).

Південна Корея сильно постраждала від раннього спалаху захворювання Covid-19, але зараз, здається, розглядається у всьому світі як один з найбільш успішних взірців для наслідування того, як найкраще боротися з пандемією. Зараз країна також робить 5G та AI центральними елементами економічного «нового курсу» в епоху після коронавірусу.

Covid-19 лише прискорює амбіції Південної Кореї щодо штучного інтелекту. Торік президент Мун Чже Ін запустив Національну стратегію штучного інтелекту, зосередивши зусилля промисловості та освіти Південної Кореї на потенційних можливостях ШІ.

Випадки, перелічені щодо можливого використання ШІ в стратегічному документі, дають відповідь на деякі найбільші проблеми, з якими стикається корейське суспільство.

Росія, країна зі старінням населення та низьким рістком, яка зараз є другим найбільш підтвердженим випадком захворювання на Covid-19, також домагається

успіхів у розвитку ШІ. Російський Яндекс, ринкова капіталізація якого становить 13,18 млрд доларів, є однією з найбільших і найперспективніших місцевих російських компаній, поєднання Google, Amazon і Uber, об'єднаних в одну, зараз є передусім компанією AI.

Найбільший російський банк, "Ощадбанк", та його генеральний директор Герман Греф у матеріалі, присвяченому "Форбс Росія", заявили, що аналітики повинні вважати його акції більше схожими на компанію з штучним інтелектом, а не просто на старомодний роздрібний банк. Ощадбанк створив альянс ШІ, запустив суперкомп'ютер і зібрав 2 мільярди доларів для російських розробників ШІ. Наприкінці минулого року Ощадбанк також співпрацював з Microsoft Research (MSR), щоб дослідити використання рішень Microsoft AI у робототехніці.

Банки за визначенням мають багато даних про клієнтів, але багато з них все ще недостатньо використовують або використовують ці дані. Глибоке навчання та ШІ можуть це перетворити.

Стенфордський професор Ендрю Нг знаменито заявив: "Дані - це нова нафта, ШІ - нова електроенергія". У світі після Covid-19 це може стати новою мантрою.

Кожен раз, коли відбувається серйозні технологічні зриви, такі як ШІ, це дає нам шанс змінити світ. ШІ - це дуже передова технологія, яка впливає як на розвинуту економіку, так і на економіку, що розвивається. Порада для економік, що розвиваються, полягає у тому, щоб зосередитись на ШІ для зміцнення вертикальних галузей країни та інвестування в освіту, оскільки ШІ все ще настільки незрілий.

У новому світі праці навчитись ефективно оцінювати, що машини можуть, а що не можна робити, а також те, що вони повинні і не повинні робити, буде критично важливим для працівників та керівників. Натомість людські сили та можливості повинні бути інтегровані в системи штучного інтелекту, щоб допомогти працівникам більш співпрацювати з ними.

Лауреат Нобелівської премії Даніель Канеманн висловив думку, що для розуміння людської поведінки шум - випадковість - навіть важливіший за "упередження", заявляючи, що він з нетерпінням чекає дня, коли штучний інтелект усуне нашу природну дурість.

Білл Гейтс був одним з небагатьох публічних голосів, що попереджав не бути самовдоволенням щодо ризику пандемії. Гейтс також заявив, що "прорив у машинному навчанні коштував би 10 мікропрограм".

Існує ще багато можливостей для кожної країни, компанії та окремої людини дізнатися більше про штучний інтелект та підготуватися, але відсутність

підготовки та самовдоволення щодо ШІ, як самовдоволення щодо коронавірусу, може бути смертельною помилкою.

## РОЗДІЛ 3 РОЗРОБКА НЕЙРОННОЇ МЕРЕЖИ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ДОСЛІДЖЕННЯ ТА АНАЛІЗУ COVID-19

### 3.1 Проектування нейронної мережи-перцептронну

Нехай, треба створити нейронну мережу, яка буде відтворювати графік, в якому буде відображатися кількість нових випадків захворювань за день, на протязі одного року (від 30 травня 2020 р. до 30 травня 2021 р. відповідно). Для цього потрібні справжні статистичні дані захворювань COVID-19, які можна подивитись на завантажити на сайті <https://ourworldindata.org/covid-cases>, на сторінці про підтверджені випадки (рисунок 3.1)

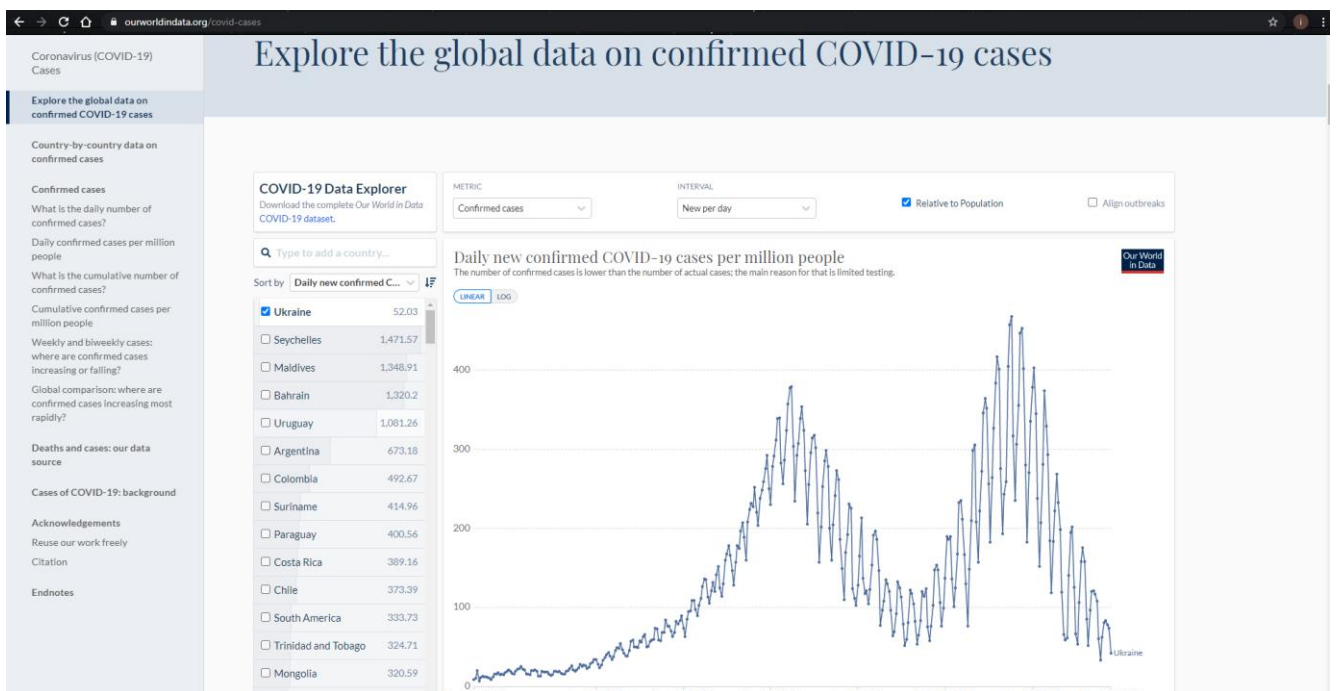


Рисунок 3.1 - Сторінка сайту для дослідження глобальних даних про підтверджені випадки захворювання на COVID-19

Було завантажено статистичні дані про підтвердженні випадки захворювань п'яти країн, таких як: Росія, Україна, Білорусь, Румунія та Болгарія. Ці дані були відсортовані в файл з розширенням електронної таблиці, з метою їх подальшого використання.

Сама нейронна мережа написана на мові програмуванні Python, для цього використовувалась середовище розробки PyCharm. PyCharm - це спеціальне інтегроване середовище розробки Python (IDE), що забезпечує широкий спектр важливих інструментів для розробників Python, тісно інтегрованих для створення зручного середовища для продуктивного розвитку Python, інтернету та науки про дані. Все, що ви робите в PyCharm, ви робите в контексті проекту. Він служить основою для допомоги у кодуванні, масового рефакторингу, узгодженості стилів кодування, тощо. Загальні налаштування застосовуються до всіх проектів певної інсталяції або версії PyCharm. Такі налаштування включають зовнішній вигляд IDE (теми, кольорові схеми, меню та панелі інструментів), налаштування сповіщень, набір встановлених та ввімкнених плагінів, налаштування налагоджувача, заповнення коду тощо.

Для нашої нейронної мережі було вирішено використовувати перцептрон, його деталі можна подивитися на рисунку 3.2.

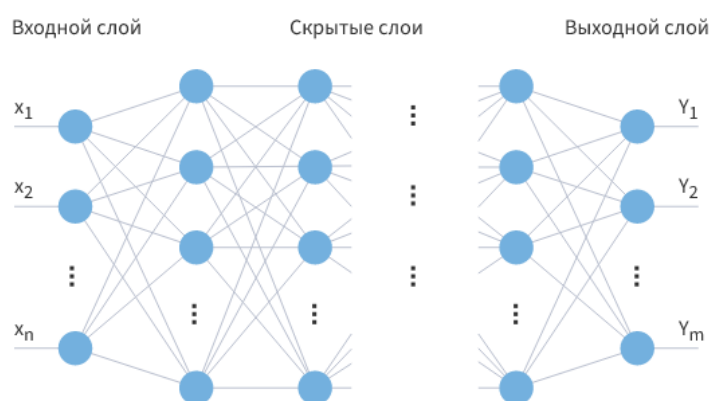


Рисунок 3.2 - Багатошаровий перцептрон

Взгалаї, перцептрон - це математична або комп'ютерна модель сприйняття інформації мозком (кібернетична модель мозку), запропонована Френком Розенблатом в 1957 році й реалізована у вигляді електронної машини «Марк-1» у 1960 році. Перцептрон став однією з перших моделей нейромереж, а «Марк-1» - першим у світі нейрокомп'ютером. Незважаючи на свою простоту, перцептрон здатен навчатися і розв'язувати досить складні завдання. Основна математична задача, з якою він здатний впоратися — це лінійне розділення довільних нелінійних множин, так зване забезпечення лінійної сепарабельності.

Перцептрон складається з трьох типів елементів, а саме: сигнали, що надходять від давачів, передаються до асоціативних елементів, а відтак до реагуючих. Таким чином, перцептрони дозволяють створити набір «асоціацій» між вхідними стимулами та необхідною реакцією на виході. В біологічному плані



це відповідає перетворенню, наприклад, зорової інформації у фізіологічну відповідь рухових нейронів. Відповідно до сучасної термінології, перцептрони може бути класифіковано як штучні нейронні мережі:

- з одним прихованим шаром;
- з пороговою передавальною функцією;
- з прямим розповсюдженням сигналу.

Існують різні види штучних нейронних мереж, що відрізняються один від іншого топологією та процесом навчання мережі. Однією з найбільш поширених і успішно застосованих у прогнозуванні топологій є багатошаровий перцептрон - мережа, в якій нейрони об'єднані в шари, причому вихідні значення, отримані з нейронами попереднього слова, що надходять у якості вхідних сигналів на нейронах наступного слова. На входи елементів першого шару мережі поступають зовнішні дані розв'язуваної задачі, а з виходами останнього шару отримують шуканий результат. Підбором вагових коефіцієнтів нейронна мережа навчається для найкращого вирішення поставленого завдання.

Багатошаровий перцептрон відноситься до класу мереж, навчання яких здійснюється методом зворотного поширення помилки. Для навчання системи використовується навчальна вибірка, що складається з навчальних прикладів, формування яких ґрунтується на вихідних даних і специфіці розв'язуваної задачі. Так, для екстраполяції часового ряду вихідні дані перетворюються в навчальну вибірку з використанням методу ковзних вікон. Відповідно до цього методу, кожен навчальний приклад вибірки формується з  $n$  значень часового ряду, званого «вікном», при цьому черговий навчальний приклад виходить шляхом зсуву вікна на одну одиницю часу.

При вирішенні задачі прогнозування система нейронної мережі будується в такий спосіб: вхідний шар містить кілька нейронів, на які подаються значення досліджуваного часового ряду, а останній шар складається з єдиного нейрона, на виході якого виходить прогноз.

Недоліком реалізації цього алгоритму є досить швидке накопичення помилки.

В результаті проведених досліджень розроблено та апробовано рішення, яке має на меті усунення описаного вище недоліку з метою збільшення точності прогнозу. Згідно з отриманими результатами запропоновано в архітектуру прогнозуючої нейронної мережі внести наступні зміни:

- збільшити кількість нейронів вихідного шару, яке визначається кількістю кроків прогнозування;
- ввести зв'язки між нейронами вихідного шару.

Збільшення точності прогнозу відбувається за рахунок зв'язування вихідних нейронів між собою так, щоб отримане значення на першому вихідному нейроні подавалося на вхід другого вихідного нейрона, а отримане на першому і другому - враховувалося на третьому і т.д. Іншими словами, на кожен наступний нейрон вихідного шару, крім сигналів з нейронів передостаннього шару, повинні подаватися сигнали, вже отримані на попередніх виходах мережі.

Згідно результатів спостережень, нейрони не реагують одразу, а подавляють вхідний сигнал до тих пір, поки він не досягне такої величини, яка запустить генерацію вихідного сигналу. Ви можете сприймати це як поріг, який повинен бути досягнутий перед тим, як буде вироблено будь-який результат. Це як вода в чашці, вода не розливається доти, доки вона спочатку не наповнить чашку. Інтуїтивно це має сенс, що нейрони не хочуть передавати крихітні шумові сигнали, лише підкреслено сильні навмисні сигнали. Це можна побачити на рисунку 3.3:

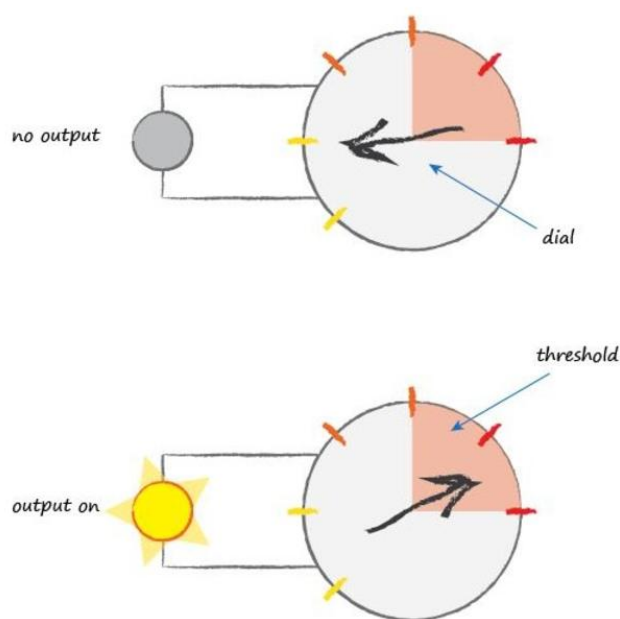


Рисунок 3.3 - Поріг збудження нейрону

Функція, яка приймає вхідний сигнал і генерує вихідний сигнал, але враховує якийсь поріг, називається функцією активації. Математично існує багато таких функцій активації, які можуть досягти цього ефекту. Проста крокова функція може це зробити:

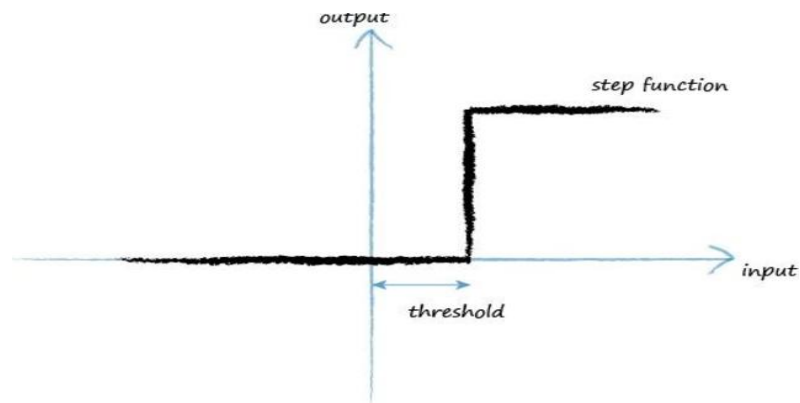


Рисунок 3.4 - Порогова функція активації

Для нашої нейронної мережі використовувалась сигмоїдна функція активації, іноді її називають логістичною функцією:

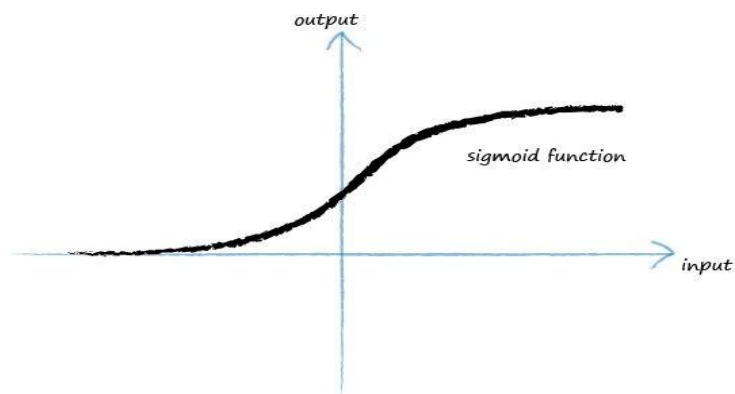


Рисунок 3.5 - Сигмоїдна функція активації

Перше, що слід усвідомити, це те, що справжні біологічні нейрони приймають багато вхідних даних, а не лише один. Що ми робимо з усіма цими входами? Ми просто поєднуємо їх, додаючи, і отримана сума є входом у функцію сигмоїди яка контролює вихід. Це відображає, як працюють справжні нейрони. Наступна діаграма ілюструє цю ідею поєднання входів, а потім застосування порогового значення до загальної суми:

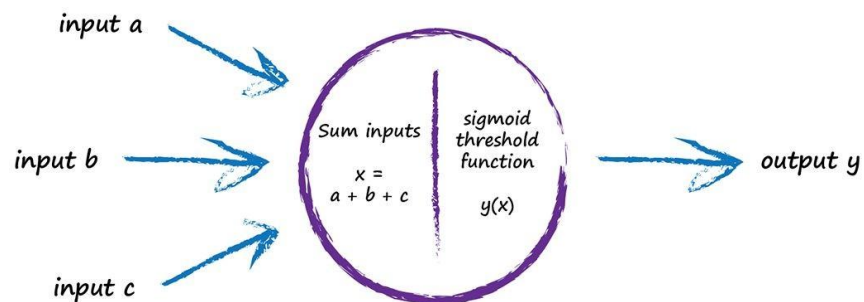


Рисунок 3.6 - Демонстрація роботи функції активації

Якщо комбінований сигнал недостатньо великий, тоді ефект порогової функції сигмоїди полягає у придушенні вихідного сигналу. Якщо сума  $x$ , якщо достатньо велика, ефект сигмоїди полягає у спрацьовуванні нейрона. Цікаво, що якщо лише один із кількох входів великий, а решта малий, цього може бути достатньо для запуску нейрона. Більш того, нейрон може спрацьовувати, якщо деякі входи є окремо майже, але не зовсім, досить великими, оскільки при комбінуванні сигнал достатньо великий, щоб подолати поріг. У інтуїтивному розумінні це дає вам відчуття більш складних і, в певному сенсі, нечітких розрахунків, які можуть робити такі нейрони.

Потрібно помітити, що кожен нейрон приймає вхідні дані від багатьох до нього, а також подає сигнали багатьом іншим, якщо трапляється, що він стріляє.

Одним із способів реплікації цього з природи на штучну модель є наявність шарів нейронів, кожен з яких зв'язаний з кожним другим у попередньому та наступному шарі. Наступний рисунок 3.7 ілюструє цю ідею:

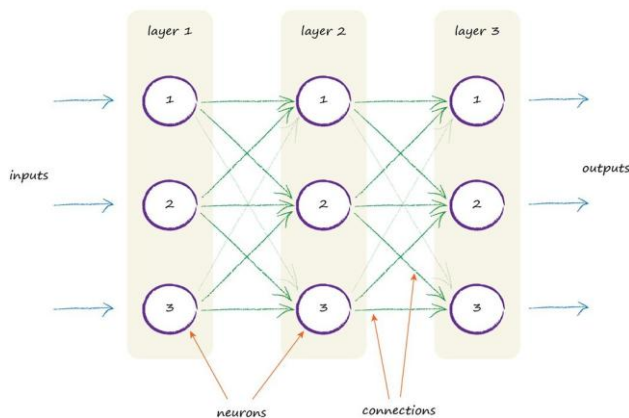


Рисунок 3.7 - Тришарова мережа

Ви можете бачити три шари, кожен із трьох штучних нейронів або вузлів. Ви також можете побачити кожен вузол, підключений до кожного іншого вузла на попередньому та наступному шарах.

Це чудово! Але яку частину цієї круто виглядаючої архітектури робить навчання? Що ми коригуємо у відповідь на навчальні приклади? Чи існує параметр, який ми можемо вдосконалити, як нахил лінійних класифікаторів, які ми розглядали раніше?

Найбільш очевидним є регулювання міцності зв'язків між вузлами. У середині вузла ми могли б скорегувати підсумовування входів або скоригувати форму порогової функції сигмоїда, але це складніше, ніж просто регулювати міцність зв'язків між вузлами.

Якщо працює більш простий підхід, дотримуйтесь його! На наступному рисунку 3.8 знову показані підключені вузли, але цього разу показано вагу,

пов'язану з кожним з'єднанням. Низька вага буде послабляти сигнал, а велика вага посилюватиме його.

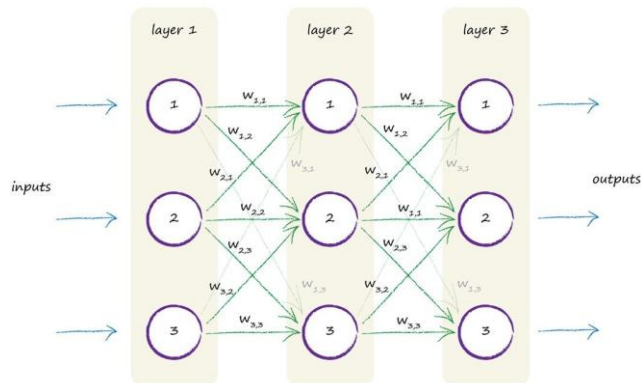


Рисунок 3.7 - Тришарова мережа з ваговими коефіцієнтами

В нашому випадку нейронна мережа складається всього з трьох шарів: вхідний, прихований та вихідний шари. Я протестував декілька конфігурацій нейронної мережі і вирішив, що для нашої задачі тришарової нейронної мережі буде достатньо, також, виявилось, що й кількості вузлів прихованого шару не потрібно надто багато, 50 вузлів буде достатньо, якщо порівнювати зі 100 вузлами, результат виходив той самий, можливо, сто вузлів мали трохи кращий результат, але його не помітиш, якщо не робити більш детальне спостереження, зате зі п'ятдесятьма вузлами процес навчання відбувався швидше.

### 3.2 Розробка та навчання нейронної мережі на Python

Для початку роботи у середовищі PyCharm потрібно встановити кілька необхідних модулів та імпортувати їх, використовуючи команду **import**, все імпортовані модулі можна подивитись на рисунку 3.9:

```
import os
import time
import sys
import matplotlib.pyplot as plt
import numpy as np
import scipy.special as sc
from PyQt5 import QtWidgets, uic
from PyQt5.QtWidgets import QApplication, QMainWindow
from pyqtgraph import PlotWidget, plot
import pyqtgraph as pg
```

Рисунок 3.10

Для написання нейронної мережі я використовував переваги об'єктно-орієнтованого програмування, що дає можливість створити клас нейронної мережі.

Об'єктно-орієнтоване програмування — це метод програмування, заснований на поданні програми як сукупності взаємодіючих об'єктів, кожен з яких є примірником певного класу, а класи є членами певної ієрархії наслідування. Програмісти спочатку пишуть клас, а на його основі під час виконання програми створюються конкретні об'єкти (екземпляри класів). На основі класів можна створювати нові, які розширюють базовий клас і таким чином створюється ієрархія класів.

На думку Алана Кея, розробника мови Smalltalk, якого вважають одним з «батьків-засновників» ООП, об'єктно-орієнтований підхід полягає в наступному наборі основних принципів:

Все є об'єктами.

Всі дії та розрахунки виконуються шляхом взаємодії (обміну даними) між об'єктами, під час якої один об'єкт потребує, щоб інший об'єкт виконав деяку дію. Об'єкти взаємодіють, надсилаючи і отримуючи повідомлення. Повідомлення — це запит на виконання дії, доповнений набором аргументів, які можуть знадобитися під час виконання дії.

Кожен об'єкт має незалежну пам'ять, яка складається з інших об'єктів.

Кожен об'єкт є представником (екземпляром, примірником) класу, який виражає загальні властивості об'єктів.

У класі задається поведінка (функціональність) об'єкта. Таким чином усі об'єкти, які є екземплярами одного класу, можуть виконувати одні й ті ж самі дії.

Класи організовані у єдину деревоподібну структуру з загальним корінням, яка називається ієрархією успадкування. Пам'ять та поведінка, зв'язані з екземплярами деякого класу, автоматично доступні будь-якому класу, розташованому нижче в ієрархічному дереві.

Таким чином, програма є набором об'єктів, що мають стан та поведінку. Об'єкти взаємодіють використовуючи повідомлення. Будується ієрархія об'єктів: програма в цілому — це об'єкт, для виконання своїх функцій вона звертається до об'єктів що містяться у ньому, які у свою чергу виконують запит шляхом звернення до інших об'єктів програми. Звісно, щоб уникнути нескінченної рекурсії у зверненнях, на якомусь етапі об'єкт трансформує запит у повідомлення до стандартних системних об'єктів, що даються мовою та середовищем програмування. Стійкість та керованість системи забезпечуються за рахунок

чіткого розподілу відповідальності об'єктів (за кожну дію відповідає певний об'єкт), однозначного означення інтерфейсів міжоб'єктної взаємодії та повної ізольованості внутрішньої структури об'єкта від зовнішнього середовища (інкапсуляції).

**class *NeuralNetwork*:** - ім'я мого класу нейронної мережі. В класі нейронної мережі я написав наступні елементи: метод (функція) конструктора, для якого ми вказуємо вхідні параметри, які буду використовуватись для навчання нейронної мережі:

- inputnodes*
- hiddennodes*
- outputnodes*
- learningrate*

*inputnodes* – кількість вхідних вузлів, *hiddennodes*, *outputnodes* та *learningrate* - кількість вузлів прихованого, вихідного шару та коефіцієнт навчання відповідно, який призначений для згладженої зміни вагів. На рисунку 3.11 можна побачити цей метод:

```
class NeuralNetwork:
    def __init__(self, inputnodes, hiddennodes, outputnodes, learningrate):
        self.inodes = inputnodes
        self.hnodes = hiddennodes
        self.onodes = outputnodes

        self.wih = np.random.normal(0.0, pow(self.hnodes, - 0.5), (self.hnodes, self.inodes))
        self.who = np.random.normal(0.0, pow(self.onodes, - 0.5), (self.onodes, self.hnodes))
        # learning coefficient
        self.lr = learningrate

        # activation function
        self.activation_function = lambda x: sc.expit(x)
        self.inverse_activation_function = lambda x: sc.logit(x)
```

Рисунок 3.11 — Метод конструктора

Також в цьому методі ми можемо побачити створення матриць (масивів) вагових коефіцієнтів,

```
self.wih = np.random.normal(0.0, pow(self.hnodes, - 0.5), (self.hnodes, self.inodes))
self.who = np.random.normal(0.0, pow(self.onodes, - 0.5), (self.onodes, self.hnodes))
```

Елементи (ваги) в матриці *wih* та *who* приймають випадкові значення вибірки з нормального (Гауссового) розподілу. Також встановлюється розмір цих матриць. Для матриці *wih*, яка містить ваги, що з'єднують між собою вузли вхідного та прихованого шарів, розмір становитиме добуток кількості вузлів прихованого на кількість вузлів вхідного шарів. А для матриці *who*, яка містить

ваги, що з'єднують між собою вузли прихованого та вихідного шарів, розмір становитиме добутку кількості вузлів вихідного на кількість вузлів прихованого шарів відповідно.

Наступною йде та сама функція-активація сигмоїда: *activation\_function = lambda x: sc.expit(x)* та обернена функція-активація: *inverse\_activation\_function = lambda x: sc.logit(x)*

Далі написано метод для тренування нейронної мережі, який можна подивитися на рисунку 3.12:

```
def train(self, inputs_list, targets_list):
    # Преобразовать список входных значений в двухмерный массив
    inputs = np.array(inputs_list, ndmin=2).T
    targets = np.array(targets_list, ndmin=2).T
    # Расчитать входящие сигналы для скрытого слоя
    hidden_inputs = np.dot(self.wih, inputs)
    # Расчитать исходящие сигналы для скрытого слоя
    hidden_outputs = self.activation_function(hidden_inputs)
    # Расчитать входящие сигналы для выходного слоя
    final_inputs = np.dot(self.who, hidden_outputs)
    # Расчитать исходящие сигналы для выходного слоя
    final_outputs = self.activation_function(final_inputs)
    # ошибка = целевое значение - фактическое значение
    output_errors = targets - final_outputs
    # ошибки скрытого слоя - это ошибки outputs, распределенные пропорционально весовым коэффициентам связей
    # и рекомбинированные на скрытых узлах
    hidden_errors = np.dot(self.who.T, output_errors)
    # обновить весовые коэффициенты связей между скрытым и выходным слоями
    self.who += self.lr * np.dot((output_errors * final_outputs * (1.0 - final_outputs)), np.transpose(hidden_outputs))
    # обновить весовые коэффициенты связей между входным и скрытым слоями
    self.wih += self.lr * np.dot((hidden_errors * hidden_outputs * (1.0 - hidden_outputs)), np.transpose(inputs))
    pass
```

Рисунок 3.12 — Метод тренування нейронної мережі

Даний метод має два вхідних параметри, список вхідних даних та список істинних значень. Список істинних значень використовується для порівняння результатів, які видає нейронна мережа, із результатами, які повинні бути. Саме в цьому методі тренування нейронної мережі написано, як саме вона навчатиметься. Тут описано обернене розповсюдження помилки, множення між матрицями вагів та зміна їх коефіцієнтів.

$$\mathit{self.who} += \mathit{self.lr} * \mathit{np.dot}((\mathit{output\_errors} * \mathit{final\_outputs} * (1.0 - \mathit{final\_outputs})), \mathit{np.transpose}(\mathit{hidden\_outputs}))$$
$$\mathit{self.wih} += \mathit{self.lr} * \mathit{np.dot}((\mathit{hidden\_errors} * \mathit{hidden\_outputs} * (1.0 - \mathit{hidden\_outputs})), \mathit{np.transpose}(\mathit{inputs}))$$

Також тут можна побачити реалізацію коефіцієнта згладжування та перемноження матриць, що впливає на зміну вагового коефіцієнту.

Наступним кроком було створення методу запиту, його можна подивитися на рисунку 3.13:



```

def query(self, inputs_list):
    inputs = np.array(inputs_list, ndmin=2).T
    # Расчитать входящие сигналы для скрытого слоя
    hidden_inputs = np.dot(self.wih, inputs)
    # Расчитать исходящие сигналы для скрытого слоя
    hidden_outputs = self.activation_function(hidden_inputs)
    # Расчитать входящие сигналы для выходного слоя
    final_inputs = np.dot(self.who, hidden_outputs)
    # Расчитать исходящие сигналы для выходного слоя
    final_outputs = self.activation_function(final_inputs)
    return final_outputs

```

Рисунок 3.13 — Метод запиту

В цьому методі описується як саме передаватимуться вхідні значення до вихідного шару; які з ними відбуваються перетворення та отримання результатів на виході. Можна привести в приклад реалізацію функції-активації, яка віддає вже вихідне значення від вузла:

*hidden\_outputs* = self.activation\_function(*hidden\_inputs*)

*final\_outputs* = self.activation\_function(*final\_inputs*)

*hidden\_inputs* — значення, які поступають до прихованого шару, *hidden\_outputs* — значення, які вже виходять з нього. *final\_outputs* — результуючі значення. До речі, цей метод запиту повертає результуючі значення.

Звернімо увагу, що для навчання нейронної мережі потрібно масштабувати всі вхідні значення, та значення для тренування до таких: від 0.01 до 0.99. Це пояснюється тим, що, якщо вхідне значення буде 0, то вузол не зможе його передати, бо незалежно від вагового коефіцієнту, буде відбуватися множення на нуль, а сигмоїдна функція-активація має область визначення, які лише наближається до 0 та 1. Тому я написав функції, які можна подивитися на рисунку 3.14, які перетворюватимуть (або масштабують) вхідні значення до інтервалу від 0.01 до 0.99:

```

def square(s):
    return (s / 17130000.0 * 0.99) + 0.01

def density(d):
    return (d / 80.3 * 0.99) + 0.01

def border(b):
    return (b / 14.0 * 0.99) + 0.01

def case(c):
    return (c / 30000.0 * 0.99) + 0.01

```

Рисунок 3.14 — функції масштабування

Далі я написав передачу необхідних файлів у програму, а саме таблиці відсортованих значень, які я завантажував:

```

cwd = os.getcwd()
first_ui_form = os.path.join(cwd, "first.ui")
all_data_csv = os.path.join(cwd, "all_data.csv")
all_data_cases_csv = os.path.join(cwd, "all_data_cases.csv")

all_data_file = open(all_data_csv, 'r')
all_data_list = all_data_file.readlines()
all_data_file.close()

all_data_cases_file = open(all_data_cases_csv, 'r')
all_data_cases = all_data_cases_file.readlines()
all_data_cases_file.close()

```

Рисунок 3.15 — передача даних у програму

На наступному рисунку 3.16 вже видно ініціалізацію нейронної мережі через метод конструктор та саме навчання нейронної мережі з використанням методу тренування:

```

def on_click_create():
    global epoch_number
    epoch_number = 0
    net._init_(form.spinBox_innodes.value(), form.spinBox_hnodes.value(), form.spinBox_onodes.value(), form.doubleSpinBox_lr.value())

def on_click_train():
    global epoch_number
    form.progressBar.setValue(0)
    form.progressBar.setMaximum(int(form.spinBox_k.value()) * int(form.spinBox_epoch.value()))
    k = 0
    count = 0
    while k < int(form.spinBox_k.value()):
        k += 1
        epochs = int(form.spinBox_epoch.value())
        for e in range(epochs):
            country_count = 0
            while country_count < count_of_countries:
                country_type = np.random.randint(0, 4)
                for day in range(366):
                    values = [((day + 1) / 366.0 * 0.99) + 0.01, square(sqrs[country_type]), density(dens[country_type]), border(brs[country_type])]
                    inputs = (np.asarray(values))
                    net.train(inputs, case(country_cases[country_type][day]))
                country_count += 1
            epoch_number += 1
            count += 1
            form.progressBar.setValue(count)
            if form.radioButton.isChecked():
                on_click_view()
        form.label_count_of_epoch.setText("Епоха: " + str(epoch_number))

```

Рисунок 3.16 — Реалізація методу тренування та навчання нейронної мережі

Саме навчання полягає в тому, що я надаю нейронній мережі чотири вхідних параметра, а саме: день, площа країни, густина населення та кількість країн, з якими межує поточна країна. Взагалі я використовую 5 країн, тому тренування полягає в тому, що для нейронної мережі надається набір вхідних даних одної із 5 країн, значення дня змінюється від 1 до 366 (1 рік), а інші три вхідні параметри країни залишаються, бо вони константи (площа, густина, к-сть межованих країн) і порівнюється з істинними значеннями, тобто новими за день підтвердженими випадками захворювання COVID-19.

Кінцевою частиною написання нейронної мережі являється вивід зображення, графіку, на основі результатів, яким навчилася сама нейронна мережа:

```

def on_click_view():
    global epoch_number
    country_type = 0
    cases = []
    if form.comboBox_country_type.currentText() == "Росія":
        country_type = 0
    elif form.comboBox_country_type.currentText() == "Україна":
        country_type = 1
    elif form.comboBox_country_type.currentText() == "Білорусь":
        country_type = 2
    elif form.comboBox_country_type.currentText() == "Румунія":
        country_type = 3
    elif form.comboBox_country_type.currentText() == "Болгарія":
        country_type = 4

    for day in range(366):
        values = [((day + 1) / 366.0 * 0.99) + 0.01, square(sqr[s[country_type]]), density(dens[country_type]), border(brs[country_type])]
        inputs = (np.asarray(values))
        outputs = net.query(inputs)
        cases.append((outputs[0][0] - 0.01) / 0.99 * 30000)

    form.graphicsView.plot(days, cases, pen=1)

```

Рисунок 3.17 — Відтворення графіків нейронною мережею

### 3.3 Створення інтерфейсу для нейронної мережі

Для написаної нейронної мережі я створював інтерфейс через програму QT Designer, але для того, щоб мати змогу працювати з інтерфейсом спочатку треба встановити модуль **pyqt5** у PyCharm, для цього потрібно було ввести в терміналі команду: *pip install pyqt5*.

QT Designer - кроссплатформенне вільне середовище для розробки графічних інтерфейсів (GUI) програм, які використовують бібліотеку Qt. Входить до складу Qt framework.

Qt Designer дозволяє створювати графічні інтерфейси користувача за допомогою ряду інструментів. Існує панель інструментів «Панель віджетів», в якій доступні для використання елементи інтерфейсу - віджети, такі як, наприклад, “випадаюче” ComboBox, “поле введення” LineEdit, “кнопка” PushButton і багато інших. Кожен віджет має свій набір властивостей, що визначається відповідним йому класом бібліотеки Qt. Властивості віджета можуть бути змінені за допомогою «Редактора властивостей». Для кожного класу властивостей віджета існує свій спеціалізований редактор. Характерною особливістю Qt Designer є підтримка візуального редагування сигналів і слотів. Так, наприклад, можна зв'язати сигнал, що генерується з переключення стану віджета CheckBox зі слотом відповідає за доступність іншого віджета.

На початку створення інтерфейсу в нас же є вікно з віджетом **central widget**, це можна побачити на рисунку 3.18:

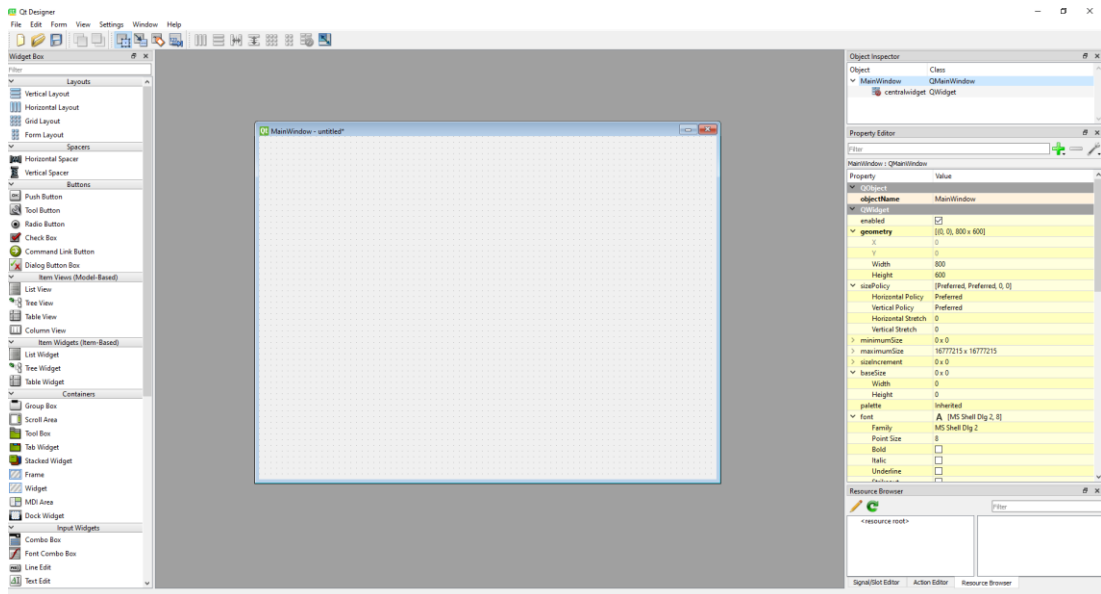
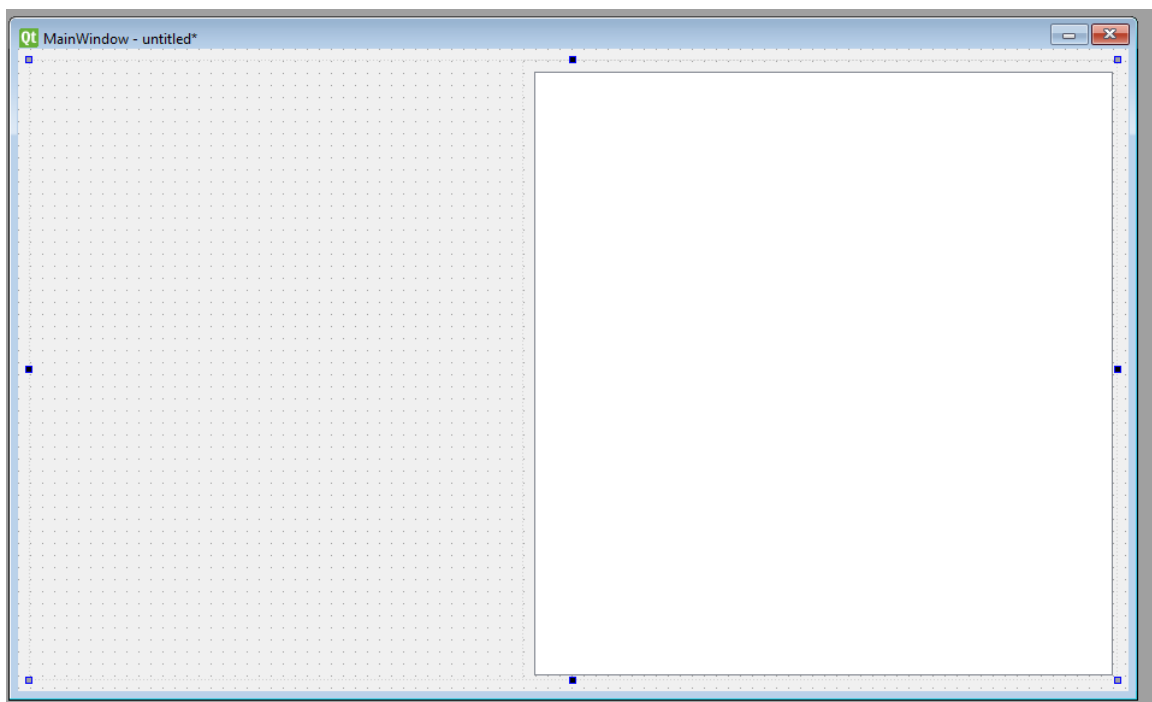


Рисунок 3.18 — Початкове вікно для створення інтерфейсу

Далі я вибрав віджет **Graphics View**, який можна побачити на рисунку 3.19, що являє собою об'єкт для роботи з графікою, саме в ньому будуть відтворюватись графіки нейронної мережі:



Р  
исун  
ок  
3.19  
—  
Відж  
ет  
Grap  
hics  
View  
для  
рабо  
ти з  
графі

КОЮ

Після цього, я додавав такі віджети, як: **Label** — віджет для відображення якого-небудь напису. **PushButton** — віджет “кнопка”, після натискання якої може відбутись якась дія, у мене в програмі є такі дії:

- Створення нейронної мережі
- Тренування нейронної мережі

Вивід графіку, яку відтворює нейронна мережа

Вивід графіку деякої країни

Очистка виведених графіків

Далі йде віджет — **ComboBox**, що являє собою випадаючий список, в якому можна вибрати певне значення, у моєму випадку я використовую його для вибору країни, для якої нейронна мережа буде відтворювати графік. **SpinBox** – віджет, що представляє собою числове поле, в якому можна вибрати певне числове значення шляхом натискання на стрілки “вниз” або “вгору”, або написати вручну. **CheckBox** – віджет, який визначає, чи стоїть “галка”, його я використовую для автовивіду графіку нейронної мережі. Скомпонувавши всі вище перераховані віджети та встановивши між ними залежність при масштабуванні вікна, я отримав функціональний інтерфейс, який можна побачити на рисунку 3.20:

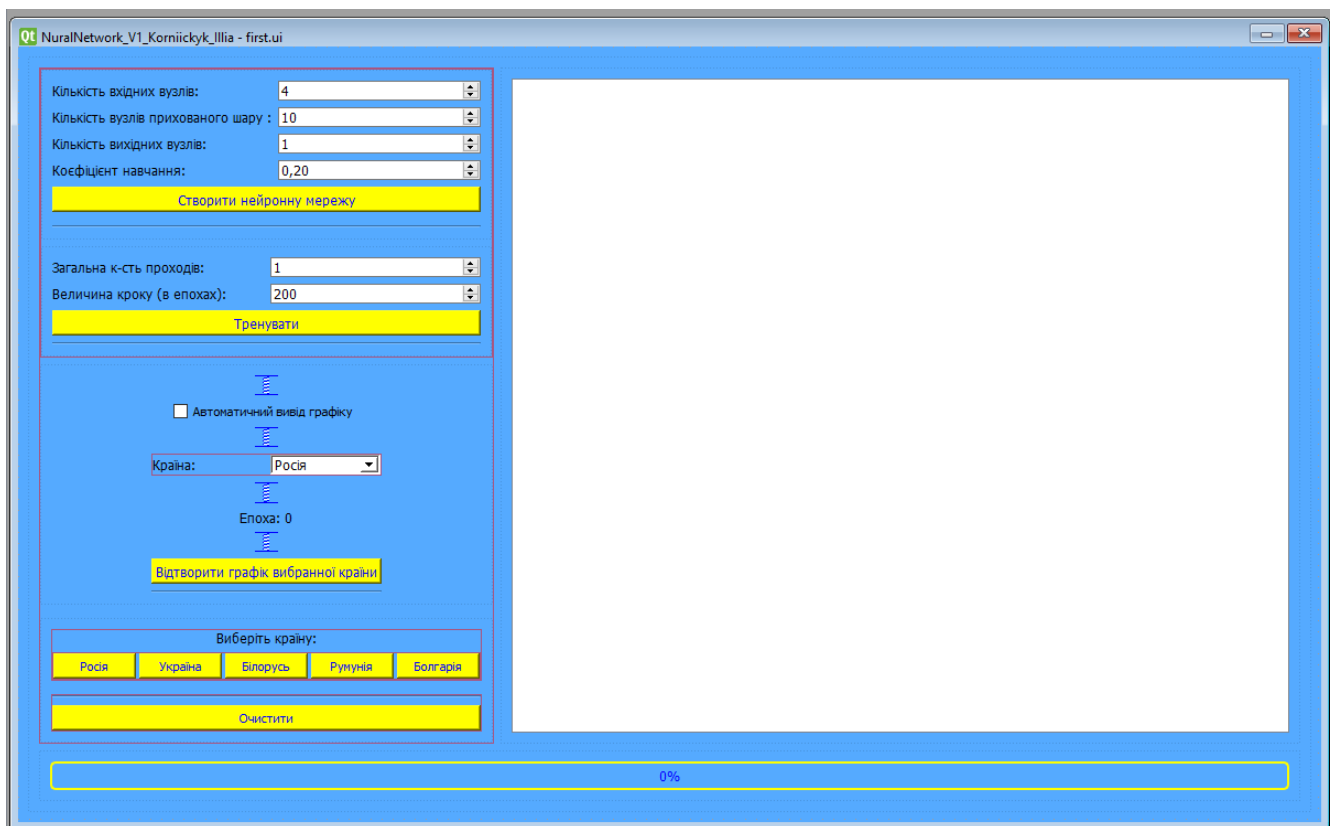


Рисунок 3.20 — Кінцевий інтерфейс в режимі редагування

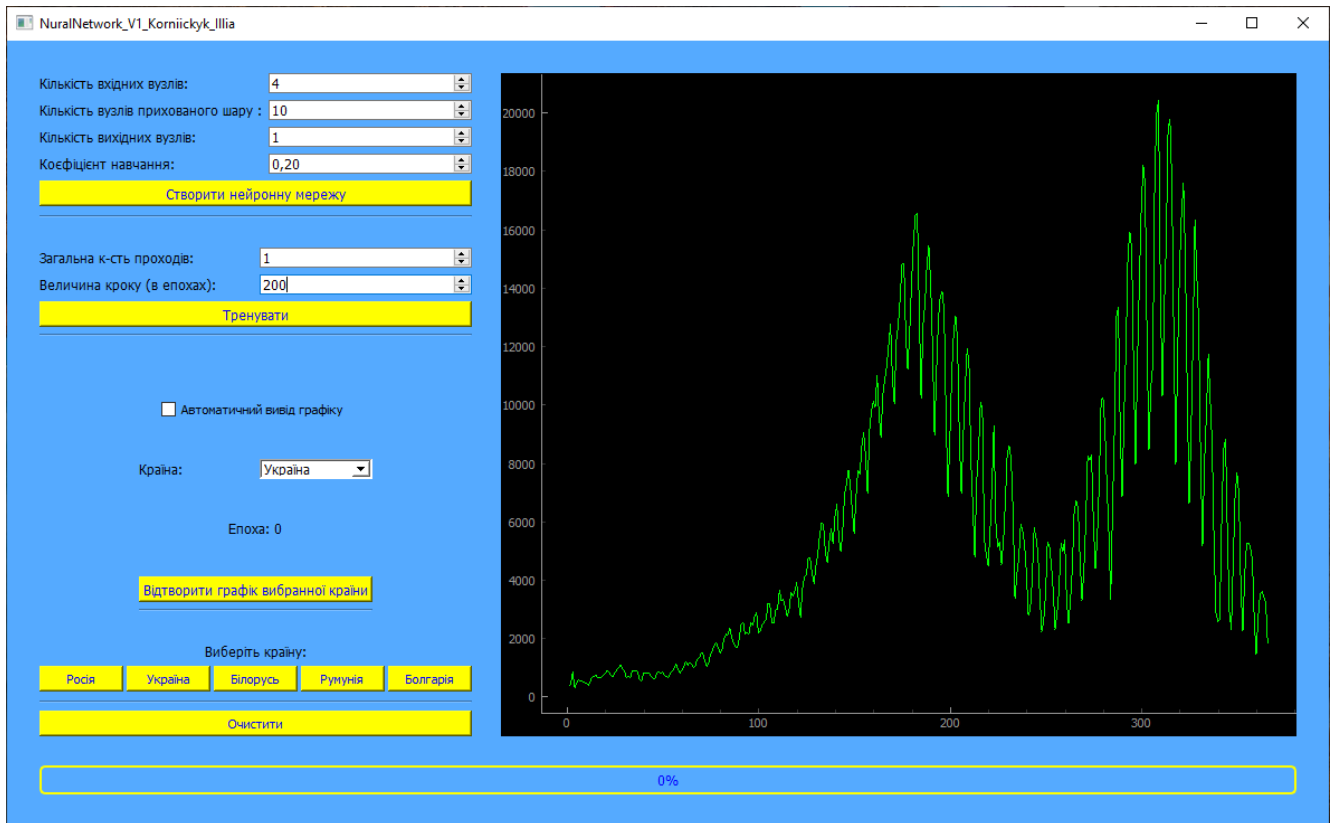


Рисунок 3.20 — Кінцевий інтерфейс в режимі користування