МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ

Звіт з переддипломної практики на тему «Рекомендаційна система для вибору ресторану за довільним запитом природною мовою за даними сервіса Yelp»

Виконав: Володько В.В.
студент 4 курсу
групи КА – 66
……

Перевірив:
керівник практики
доцент Тимощук О.Л.

Оцінка _____
"___" _____ 2020 р.

Науковий керівник дипломної роботи: асист. Макуха М.П.
рекомендовано оцінка: _____

Підпис

Київ 2020

Зміст

Вступ

Протягом значного проміжку часу в усьому світі швидкими темпами збільшується кількість інформації. Люди кожного дня сприймають та фільтрують вхідний потік інформації, що надходить з різних джерел: робота, побутові проблеми, популярні джерела інформації тощо. Після винайдення мережі Інтернет кількість такої інформації стала стрімко зростати, з'явилася велика кількість сервісів для надання користувачам всього необхідного для комфортного життя.

За останню декаду набули значної популярності інтернет-сервіси, що пропонують товари всіх можливих видів (інтернет-магазини), інформацію на будь-який смак (інтернет-журнали, новини, книги, статті) тощо. Користувачу стало надзвичайно важко орієнтуватися в каталогах товарів та списках статей, навіть із вбудованим пошуком та фільтрацією, оскільки дуже важко зробити вибір при настільки великому об'ємі інформації.

Рекомендаційні системи з'явилися на сучасному ринку IT як механізм для заміни статичному списку рекомендацій при пошуку або покупках на веб-сайтах. Ці системи формують рейтинговий перелік об'єктів (товарів, фільмів, музичних композицій) на основі різних критеріїв: релевантність, популярність, історія оцінок тощо.

Такі системи почали широко використовувати існуючі інтернет-компанії в рамках інтернет-маркетингу. За допомогою прогнозування рекомендацій вони мають на меті збільшити залученість користувачів до конкретного сервісу. Також, при розробці рекомендаційної системи з релевантними рекомендаціями, що заслужили довіру користувачів, можна розміщувати серед цих рекомендацій інші товари, що рекламуються.

Разом зі більшенням кількості рекомендацій для користувачів, які мають історію запитів появилась необхідність надавати рекомендації новим користувачам, про яких нічого не відомо, а є лише запит.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ВИДІВ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Визначення рекомендаційних систем

Рекомендаційна система – це система, що рекомендує товари користувачам серед величезного потоку інформації, в залежності від їх потреб. Товари – елементи конкретного сервісу, до яких користувач має інтерес: фільми, ресторани, книги, статті і т.ін. Інтереси користувачів можуть бути представлені декількома способами: із застосуванням оцінок, які користувачі надають товарам або за допомогою ключових слів кожного товару. [1]

Щоб зберігати вподобання користувачів стосовно товарів, РС використовують профілі користувачів. У більшості РС профіль користувача містить набори оцінок та/або ключових слів (тегів). Оцінки, надані користувачами товарам, можуть належати різним проміжкам (0-1, 1-5, 1-10): чим вищий рейтинг, тим більше конкретний товар сподобався користувачу.

Після кожного оцінювання все рейтинги користувача агрегуються через ряд обчислень, вимірюються схожість користувачів, а потім прогнозуються рекомендації для даного користувача. Ключові слова автоматично підвантажуються з текстів або товарів, які користувачі проглядали або оцінювали в минулому. Вони також можуть мати ваги в залежності від того, на скільки користувач оцінив конкретне слово, або більш значущі слова матимуть більшу вагу, ніж менш значущі (алгоритм TF-IDF). Після цього тексти (товари) зіставляються з профілем користувача та найбільш відповідающі йому – рекомендуються.

Рейтинги можуть бути явними та неявними. Явна оцінка – це оцінка, якою користувач показав зацікавленість даним товаром в межах своєї системи оцінювання. Неявні рейтинги вираховуються з історії покупок або поведінки користувачів. До їх переваг можна віднести зниження навантаження на користувача оцінюванням товарів. Джерелом неявних рейтингів можуть бути час, витрачений на читання статті, посилання на товар в інших джерелах (наприклад алгоритм ранжування сторінок Google). Інші індикатори поведінки перегляду, як рух курсору, ввід клавіатури та швидкість прокрутки сторінки, також були досліджені в якості неявних показників інтересу та показали непогані результати.

Загальна архітектура рекомендаційної системи (рис. 1.1) може бути представлена наступним чином:

1) Довідкові дані (background data) – системна інформація про товари сервісу, що збирається, як правило, в фоновому режимі.

2) Вхідні дані (input data) – інформація, яку користувач вносить в систему, щоб отримати рекомендації.

3) Рекомендаційний алгоритм - алгоритм, що комбінує системну інформацію та вхідні дані для отримання рекомендацій.

Типові системи в якості довідкових даних використовують профілі користувачів, а в якості вхідних – дії користувача (оцінювання товарів, час, проведений на сторінці, тощо). [10]

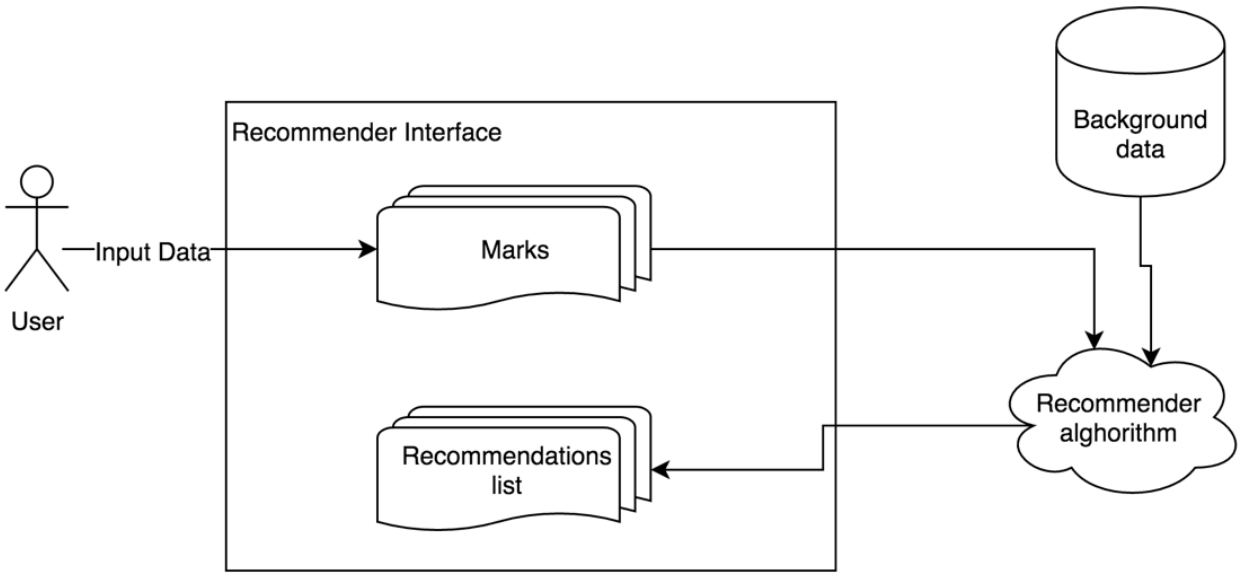

Рисунок 1.1 – Базова архітектура рекомендаційної системи

1.2 Типи підходів до створення рекомендаційних мереж

1.2.1 Спільна фільтрація

Один із підходів до проектування систем рекомендацій, який широко застосовується, - це спільна фільтрація . Спільна фільтрація базується на припущенні, що люди, які домовилися в минулому, згодяться в майбутньому, і що їм сподобаються подібні предмети, як вони сподобалися в минулому. Система генерує рекомендації, використовуючи лише інформацію про рейтингові профілі для різних користувачів або елементів. Розташовуючи однорангових користувачів / елементи, що мають історію рейтингу, аналогічну поточному користувачеві або елементу, вони генерують рекомендації, використовуючи цей мікрорайон. Методи спільної фільтрації класифікуються на основі пам’яті та моделей. Добре відомим прикладом підходів, заснованих на пам'яті, є алгоритм, заснований на користувачі, тоді як модельний підхід – це рекомендатор ядро-картографування.

Ключовою перевагою підходу спільної фільтрації є те, що він не покладається на машинно проаналізований вміст, і тому він здатний точно рекомендувати складні елементи, такі як фільми, не вимагаючи «розуміння»

самого елемента. Багато алгоритмів використовувались для вимірювання подібності користувачів або подібності елементів у системах рекомендацій. Наприклад, k-найближчий сусід (k-NN) та пірсонова кореляція, як вперше реалізований Алленом.

При побудові моделі з поведінки користувача часто розрізняють явні та неявні форми збору даних .

Підходи до спільної фільтрації часто страждають від трьох проблем: холодний старт , масштабованість та рідкість.

Холодний старт : для нового користувача або елемента недостатньо даних для точних рекомендацій.

Масштабованість : У багатьох середовищах, в яких ці системи дають рекомендації, є мільйони користувачів та продуктів. Таким чином, для обчислення рекомендацій часто потрібна велика кількість обчислювальної потужності.

Небагатості : кількість товарів, що продаються на основних сайтах електронної комерції, надзвичайно велика. Найактивніші користувачі оцінюють лише невеликий підмножину загальної бази даних. Таким чином, навіть найпопулярніші предмети мають дуже мало оцінок.

Одним з найвідоміших прикладів спільної фільтрації є спільна фільтрація по предметах (люди, які купують x, також купують y), алгоритм, популяризований системою рекомендацій Amazon.com .

1.2.2 Фільтрування на основі вмісту

Ще один поширений підхід при розробці систем рекомендацій - це фільтрація на основі вмісту . Методи фільтрування на основі вмісту засновані на описі елемента та профілі налаштувань користувача. Ці методи найкраще підходять для ситуацій, коли відомі дані про предмет (ім'я, місцезнаходження, опис тощо), але не про користувача. Рекомендатори на основі вмісту розглядають рекомендації як специфічну для користувача проблему класифікації та вивчають класифікатор лайків та сподобань користувача на основі особливостей товару.

У цій системі ключові слова використовуються для опису елементів, а профіль користувача будується для позначення типу предмета, який

подобається цьому користувачеві. Іншими словами, ці алгоритми намагаються рекомендувати предмети, схожі на ті, які сподобався користувачеві в минулому або вивчає в сьогоденні. Він не покладається на механізм входу користувача для створення цього часто тимчасового профілю. Зокрема, різні позиції кандидатів порівнюються з предметами, які раніше оцінив користувач, і рекомендується найкраще відповідати. Цей підхід має своє коріння в дослідженнях пошуку та фільтрації інформації .

Для створення профілю користувача система в основному зосереджується на двох типах інформації:

1. Модель уподобань користувача.

2. Історія взаємодії користувача з системою рекомендацій.

В основному ці методи використовують профіль елемента (тобто набір дискретних атрибутів і особливостей), що характеризують елемент у системі. Для абстрагування особливостей елементів у системі застосовується алгоритм подання елементів. Широко використовуваний алгоритм - це зображення tf – idf (його також називають векторним представленням простору). Система створює контентний профіль користувачів на основі зваженого вектора ознак елементів. Ваги позначають важливість кожної функції для користувача і можуть бути обчислені з індивідуально оцінених векторів вмісту, використовуючи різні методи. Прості підходи використовують середні значення векторного рейтингу предмета, а інші складні методи використовують методи машинного навчання, такі як Байєсові класифікатори ,аналіз кластерів , дерева рішень та штучні нейронні мережі , щоб оцінити ймовірність того, що користувачеві цей предмет сподобається.

Основна проблема фільтрування на основі вмісту полягає в тому, чи система здатна дізнатися налаштування користувачів з дій користувачів щодо одного джерела вмісту та використовувати їх для інших типів вмісту. Якщо система обмежується рекомендацією контенту того ж типу, який користувач вже використовує, значення системи рекомендацій значно менше, ніж коли можна рекомендувати інші типи вмісту з інших служб. Наприклад, рекомендувати статті новин на основі перегляду новин корисно, але було б набагато корисніше, коли музика, відео, продукти, дискусії тощо від різних служб можуть бути рекомендовані на основі перегляду новин. Для подолання

цього більшість систем, що базуються на контентних рекомендаціях, зараз використовують певну форму гібридної системи.

Системи рекомендацій на основі вмісту можуть також включати системи рекомендацій на основі думки. У деяких випадках користувачі можуть залишати огляд тексту або відгуки про елементи. Ці генеровані користувачем тексти - це неявні дані для системи рекомендацій, оскільки вони є потенційно багатим ресурсом як функцій / аспектів товару, так і оцінок / настроїв користувачів до елемента. Особливості, витягнуті з оглядів, створених користувачем, є вдосконаленими мета-даними елементів, оскільки, оскільки вони також відображають такі аспекти елемента, як метадані , користувачі, що витягуються, широко занепокоєні користувачами. Відчуття, витягнуті з відгуків, можна розглядати як оцінки користувачів за відповідні функції. У популярних підходах до системи рекомендацій, заснованих на думках, використовуються різні методи, включаючи пошук тексту, пошук інформації , аналіз настроїв (див. також Мультимодальний аналіз настроїв) та глибоке навчання.

1.2.3 Рекомендаційні системи з кількома критеріями

Системи рекомендацій із кількома критеріями (MCRS) можуть бути визначені як системи рекомендацій, що містять інформацію про перевагу за кількома критеріями. Замість того, щоб розробляти методи рекомендацій, засновані на значенні одного критерію, загальної переваги користувача u для елемента i, ці системи намагаються передбачити рейтинг для невивчених елементів u, використовуючи інформацію про переваги за кількома критеріями, які впливають на це загальне значення переваги. Кілька дослідників підходять до MCRS як багатокритеріальної проблеми прийняття рішень (MCDM) і застосовують методи та методи MCDM для впровадження систем MCRS.

1.2.4 Мобільні системи рекомендацій

Мобільні системи рекомендування використовують смартфони з доступом до Інтернету, щоб запропонувати персоналізовані, залежні від контексту рекомендації. Це особливо складний напрямок досліджень, оскільки мобільні дані складніші, ніж дані, з якими часто доводиться стикатися з системами рекомендацій. Він неоднорідний, галасливий, вимагає

просторової та часової автокореляції, має проблеми з валідацією та загальністю.

Є три фактори, які можуть впливати на системи мобільних рекомендацій та точність результатів прогнозування: контекст, метод рекомендацій та конфіденційність. Крім того, мобільні системи рекомендування страждають від проблеми з трансплантацією - рекомендації можуть застосовуватися не у всіх регіонах (наприклад, було б нерозумно рекомендувати рецепт у тому районі, де всі інгредієнти можуть бути недоступними).

Одним із прикладів системи мобільних рекомендацій є підходи таких компаній, як Uber та Lyft, щоб створити маршрути для водіїв таксі в місті. Ця система використовує GPS-дані про маршрути, які їздять таксисти під час роботи, включаючи місцезнаходження (широту та довготу), часові позначки та робочий стан (з пасажирами чи без них). Він використовує ці дані, щоб рекомендувати перелік пунктів набору по маршруту з метою оптимізації часу зайнятості та прибутку.

Мобільні системи рекомендацій також успішно побудовані з використанням "Веб-даних" як джерела структурованої інформації. Хорошим прикладом такої системи є SMARTMUSEUM Система використовує методи семантичного моделювання, пошуку інформації та машинного навчання для того, щоб рекомендувати вміст, що відповідає інтересам користувача, навіть якщо вони представлені з обмеженими або мінімальними даними користувачів.

1.2.5 Гібридні рекомендаційні системи

Більшість систем рекомендацій зараз використовують гібридний підхід, поєднуючи спільну фільтрацію, фільтрацію на основі вмісту та інші підходи. Немає жодної причини, чому кілька різних технік одного типу не можна було би гібридизувати. Гібридні підходи можуть бути реалізовані декількома способами: шляхом складання прогнозів на основі контенту та спільної роботи окремо, а потім їх поєднання; додаючи можливості на основі вмісту до підходу, що базується на співпраці (і навпаки); або об'єднавши підходи в одну модель. Кілька досліджень, які емпірично порівнюють ефективність роботи гібриду з чистими методами спільної роботи та вмісту та показали, що гібридні методи можуть дати більш точні рекомендації, ніж чисті

підходи. Ці методи також можуть бути використані для подолання деяких найпоширеніших проблем у системах рекомендацій, таких як холодний старт та проблема зрідженості, а також вузьким місцем в галузі інженерних знань у підходах на основі знань.

Деякі методи гібридизації включають:

- Зважений : Комбінування кількості різних рекомендаційних компонентів чисельно.
- Переключення : вибір між рекомендаційними компонентами та застосування обраного.
- Змішані : Рекомендації різних рекомендаторів подаються разом, щоб дати рекомендацію.
- Поєднання особливостей: Особливості, отримані з різних джерел знань, поєднуються разом і надаються одному алгоритму рекомендацій.
- Розширення функції : обчислення функції або набору функцій, яка потім є частиною вводу до наступної методики.
- Каскад : Рекомендаціям надається суворий пріоритет, причому нижчі пріоритетні порушують зв'язки в оцінці вищих.
- Метарівень : застосовується одна методика рекомендацій і виробляється якась модель, яка потім є входом, використовуваним наступною методикою.

1.2.6 Системи рекомендацій, що обізнані з ризиком

Більшість існуючих підходів до системи рекомендацій зосереджені на тому, щоб рекомендувати найбільш релевантний вміст користувачам, які використовують контекстну інформацію, але не враховують ризик заважати користувачу небажаним сповіщенням. Важливо враховувати ризик засмутити користувача, висуваючи рекомендації за певних обставин, наприклад, під час професійної зустрічі, рано вранці або пізно вночі. Тому ефективність системи рекомендацій частково залежить від ступеня, до якої вона включила ризик у процес рекомендацій. Одним із варіантів управління цією проблемою є DRARS - система, яка моделює рекомендації з урахуванням контексту як проблеми з бандитами. Ця система поєднує в собі контент-техніку та контекстний алгоритм бандитів

1.3 Холодний старт

Холодний старт в обчисленні означає проблему, коли система або її частина була створена або перезапущена і не працює при нормальній роботі. Проблема може бути пов'язана з ініціалізацією внутрішніх об'єктів або заселенням кешу або запуском підсистем.

У типових системах веб-сервісу проблема виникає після перезавантаження сервера, а також при очищенні кешу (наприклад, після виходу нової версії). Перші запити до веб-сервісу спричинить значно більше навантаження через заповнення кешу сервера та очищення кеш- пам’ятника браузера та запрошення нових ресурсів. Іншим службам, як проксі-кешування або веб-прискорювач , також буде потрібен час для збору нових ресурсів та нормальної роботи.

Аналогічна проблема виникає при створенні екземплярів у розміщеному середовищі та екземплярах служб хмарних обчислень .

Холодний старт (або холодний завантажувач) також може стосуватися процесу завантаження одного комп’ютера (або віртуальної машини). У цьому випадку служби та інші програми запуску виконуються після перезавантаження. Зазвичай система стає доступною для користувача, навіть якщо операції запуску все ще виконуються та сповільнюють інші операції.

Інший тип проблем - це коли модель даних певної системи вимагає з'єднань між об'єктами. У такому випадку нові об'єкти не працюватимуть нормально, поки ці з'єднання не будуть виконані. Це добре відома проблема із системами рекомендацій .

Висновок до розділу 1

1. Наведено дані про рекомендаційні системи
2. Наведено типи рекомендаційних систем
3. Коротко описано кожен тип рекомендаційних систем

РОЗДІЛ 2. ТЕОРЕТИЧНІ ОСНОВИ НЕЙРОННИХ МЕРЕЖ

Завдання обробки запитів природньою мовою було поставлене ще Аланом Тюрінгом на початку історії електронних обчислювальних машин та розвитку штучного інтелекту. З швидким розвитком нейронних мереж з'являються все нові та нові методи їх створення, навчання та варіанти використання.

Інтелектуальні системи на основі штучних нейронних мереж дозволяють успішно вирішувати проблеми розпізнавання образів, виконання прогнозів, оптимізації, асоціативної пам'яті та керування. Також відомі і інші підходи до вирішення цих проблем, але вони не мають такої гнучкості, як нейронні мережі. Для даного дослідження неабияке значення має властивість нейронної мережі – адаптивність. Ця властивість дозволяє отримувати прогноз, використовуючи невеликий масив даних, покращуючи його під час отримання нових даних. Саме цим обумовлений вибір штучної нейронної мережі, як інтелектуального метода для розв'язання поставленої задачі.

2.1 Розвиток досліджень в області ШНМ

Дослідження в області штучних нейронних мереж пережили три етапи розвитку. Перший етап (40-і роки XX сторіччя) пов'язують з роботою МакКаллока та Піттса [1]. Другий етап розпочався в 60-х роках завдяки теоремі збіжності персептрона Розенблатта [2] та роботі Мінського і Пейперта [3], що вказала на обмежені можливості найпростішого персептрона. Результати Мінського і Пейперта згасили ентузіазм більшості дослідників, особливо тих, хто працював в області обчислювальних наук. Затишок, який виник в дослідженнях з нейронних мереж, тривав майже 20 років.

З початку 80-х років штучні нейронні мережі знов викликали інтерес дослідників. Це було пов'язано з енергетичним підходом Хопфілда [4] та

алгоритмом зворотного розповсюдження помилки для навчання багатошарового персептрона (багатошарові мережі прямого розповсюдження), вперше запропонованого Вербосом [5] та незалежно розробленого рядом інших авторів. Широку популярність алгоритм отримав завдяки роботі Румельхарта [6].

2.2 Біологічний нейрон та його технічна модель

Нейрон являється особливою біологічною клітиною, яка обробляє інформацію (рис. 1.1). Ця клітина складається з тіла (cell body), або соми (soma), та двох типів зовнішніх деревоподібних гілок: аксона (axon) та дендритів (dendrites). Однією із складових тіла нейрона є ядро (nucleus), яке містить інформацію про спадкові властивості. Нейрон отримує сигнали (імпульси) від інших нейронів через дендрити (приймачі) та передає сигнали, згенеровані тілом клітини, вздовж аксона (передавача), який в кінці розгалужується на волокна (strands). На кінцях цих волокон знаходяться синапси (synapses).

Синапс – елементарний структурний та функціональний вузол між двома нейронами (волокна аксона одного нейрона та дендрит іншого). Коли імпульс досягає синаптичної кінцівки, виділяються певні хімічні речовини. Ці речовини дифундують через синапс, збуджуючи чи гальмуючи здатність нейрона-приймача генерувати електричні імпульси в залежності від їх типу.

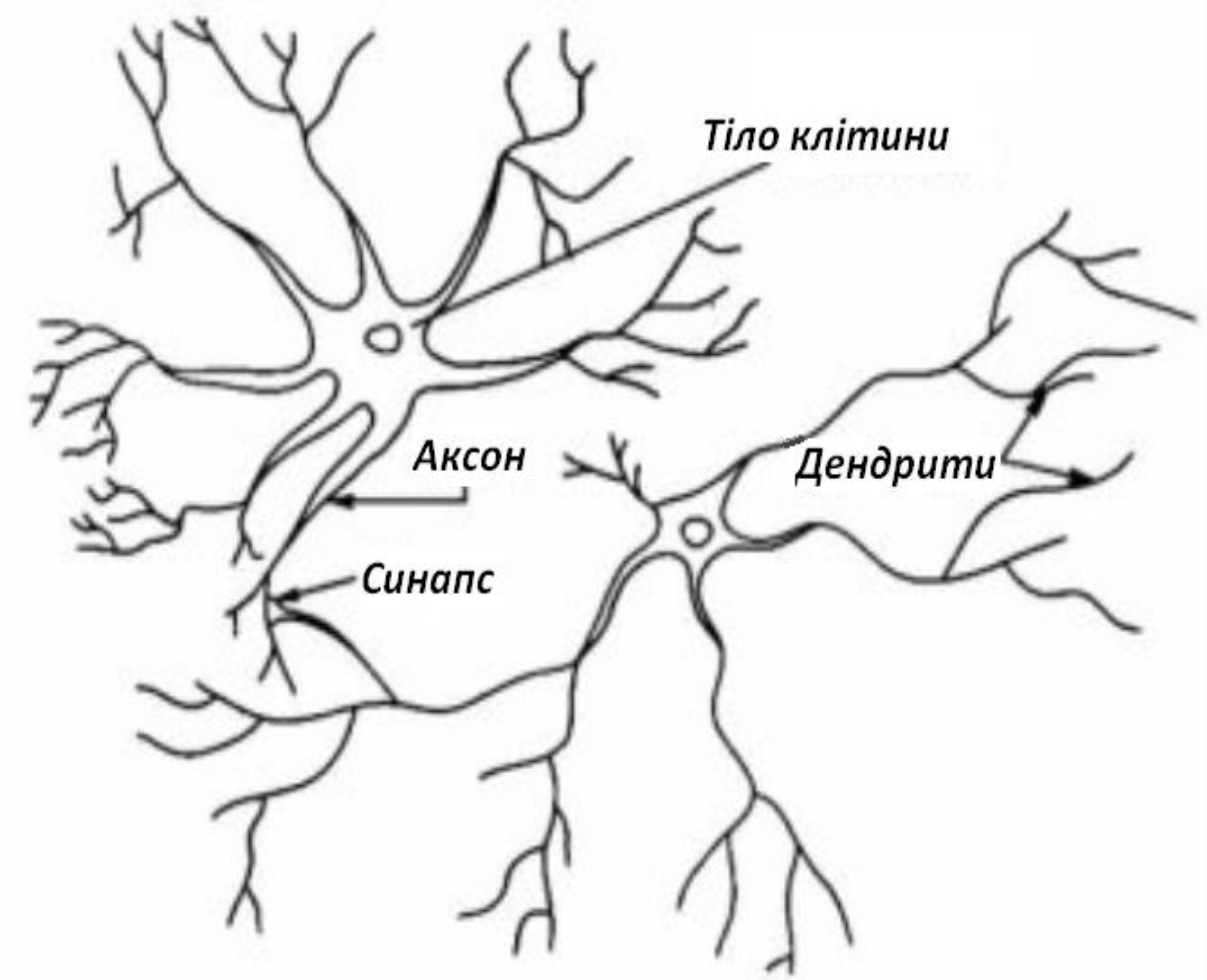

Рисунок 2.1 – Графічне зображення біологічного нейрона

Результативність синапса може налаштовуватися сигналами, що проходять через нього. Таким чином, синапси можуть вчитися в залежності від активності процесів, в яких вони приймають участь. Ця залежність від попередньої історії діє як пам'ять, яка, можливо, відповідальна за пам'ять людини.

На основі біологічного нейрона МакКаллок і Піттс [1] запропонували використовувати в якості моделі штучного нейрона бінарний пороговий елемент. Цей математичний нейрон (рис. 1.2) розраховує зважену суму n вхідних сигналів xi, i = 1, 2, …, n в блоці, позначеному Σ, і формує на виході сигнал (OUT) величини 1, якщо зважена сума перевищує встановлений поріг u, та 0 – в іншому випадку. Таким чином, штучний нейрон імітує властивості біологічних нейронів.

В багатьох випадках зручно розглядати u як ваговий коефіцієнт wi, що пов'язаний з постійним входом x0 = 1. Додатні ваги відповідають збуджуючим зв’язкам, а від’ємні – гальмівним. МакКаллок і Піттс [1] довели, що сукупність паралельно функціонуючих нейронів такого типу, при належним чином підібраних вагах, здана виконувати універсальні обчислення. Отже спостерігається певна аналогія з біологічним нейроном: передача сигналів імітує взаємодію аксонів та дендритів, ваги зв’язків відповідають синапсам, а порогова функція відображає активність соми.

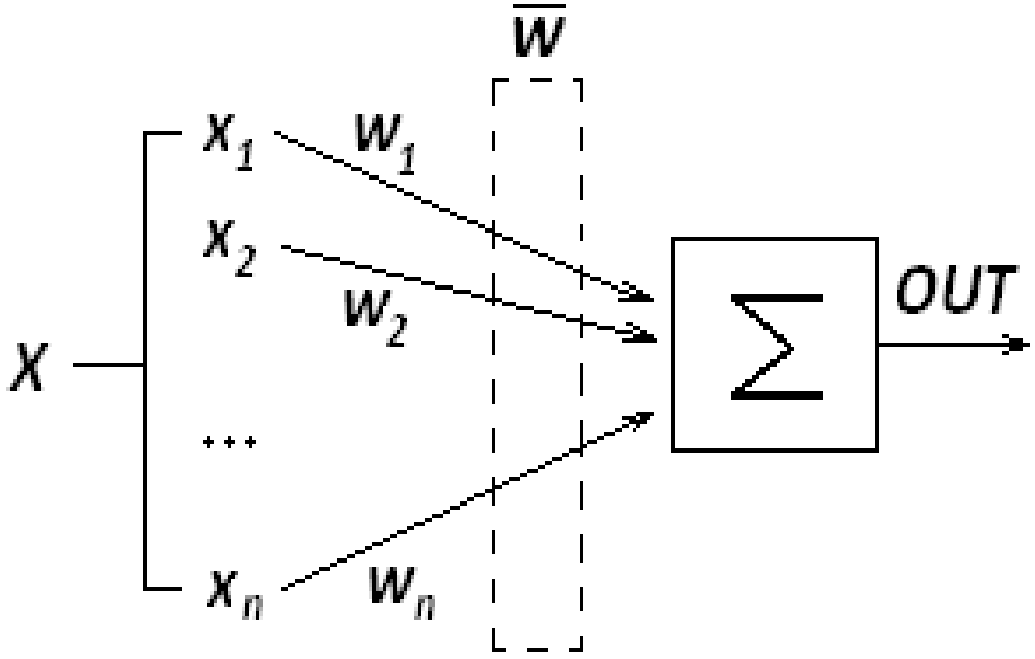

Рисунок 2.2 – *Схема штучного нейрона*

Вихідний сигнал суматора OUT (рис. 2.2) надалі, як правило, перетворюється активаційною функцією нейрона, формуючи остаточний вихідний сигнал. В якості активаційної функції, крім порогової, використовують будь-яку лінійну функцію виду f(x) = k·x або нелінійну функцію, що більш точно моделює передаточну характеристику біологічного нейрона і надає нейронній мережі більше можливостей. Найбільш уживаною активаційною функцією є логістична, або сигмоїдальна (S-подібна) функція. Математично ця функція визначається як,

$$Y = \frac{1}{1 + e^{-OUT}}, \tag{2.1}$$

де Y – вихідний сигнал нейрона.

Логістична функція дозволяє перетворити діапазон вхідного сигналу нейрона на будь-який скінчений інтервал. В цьому можна переконатись, побудувавши графік сигмоїди (рис. 1.3).

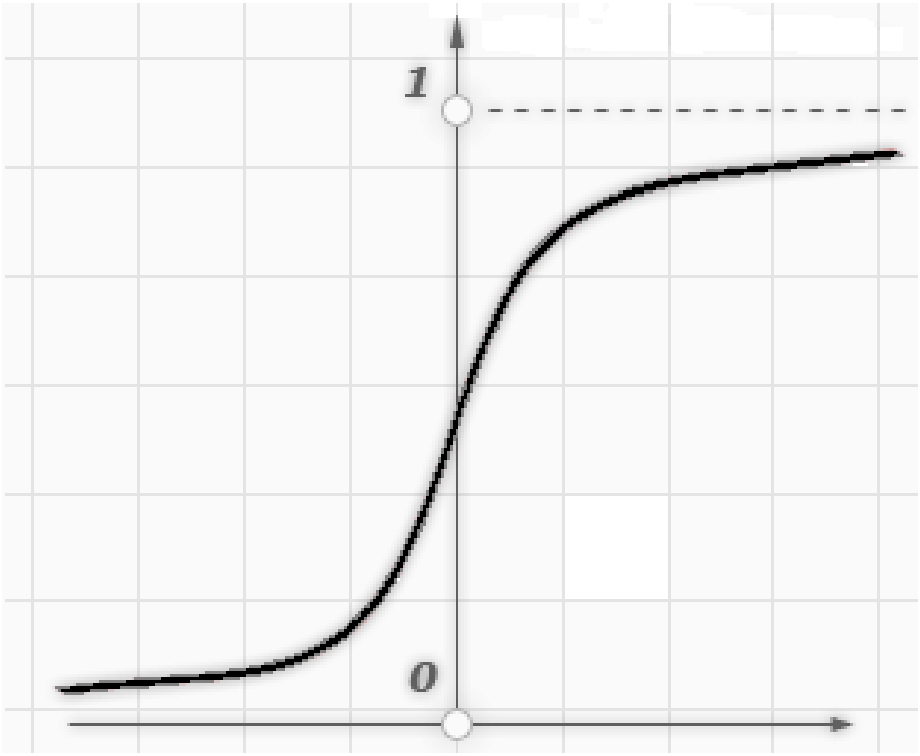

Рисунок 1.3 – *Сигмоїдальна активаційна функція*

Дану активаційну функцію можна вважати нелінійною посилюючою характеристикою штучного нейрона. Коефіцієнт посилення виражається нахилом кривої при певному рівні збудження і змінюється від малих значень при великих негативних збудженнях (крива майже горизонтальна) до максимального значення при нульовому збудженні і знову зменшується, коли збудження стає великим позитивним.

Розглянута проста модель штучного нейрона ігнорує багато властивостей свого біологічного двійника. Наприклад, вона не бере до уваги затримки в часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. І, що важливіше, вона не враховує впливів

функції частотної модуляції або синхронізуючої функції біологічного нейрона, які ряд дослідників вважають вирішальними [7].

Незважаючи на ці обмеження, мережі, побудовані з цих нейронів, виявляють властивості, які подібні до біологічної системи.

2.3 Багатошарова архітектура нейронної мережі

Нейронна мережа представляє собою сукупність нейроподібних елементів – штучних нейронів, певним чином з'єднаних між собою та з зовнішнім середовищем за допомогою зв'язків, які визначаються ваговими коефіцієнтами [8]. Штучна нейронна мережа може розглядатися, як направлений граф зі зваженими зв'язками, в якому штучні нейрони виступають вузлами. За архітектурою зв'язків штучні нейронні мережі можуть бути згруповані в два класи: мережі прямого розповсюдження, в яких графи не мають петель, та рекурентні мережі, або мережі зі зворотними зв'язками.

В найбільш розповсюдженому сімействі мереж першого класу нейрони розташовані шарами та мають однаково направлені зв'язки між шарами. Такі мережі називають багатошаровими персептронами (рис. 1.4). Багатошаровий персептрон – окремий випадок персептрона Розенблатта, в якому один алгоритм зворотного поширення помилки навчає всі шари. Назва з історичних причин не відображає особливості даного виду персептрона, тобто не пов'язана з тим, що в ньому є кілька шарів (бо кілька шарів було і у персептрона Розенблатта). Особливістю є наявність більш ніж одного шару, що навчається (як правило - два або три). Для застосування більшого числа шарів на даний момент немає обґрунтування, при цьому втрачається швидкість обчислень без придбання якості. Більш того, необхідність у великій кількості шарів-учнів відпадає, так як теоретично єдиного прихованого шару досить, щоб перекодувати вхідний сигнал таким чином, щоб отримати лінійну роздільність для вихідного подання. Існує припущення, що, використовуючи більше число шарів, можна зменшити число елементів в них, тобто сумарне число елементів в шарах буде менше, ніж якщо використовувати один прихований шар.

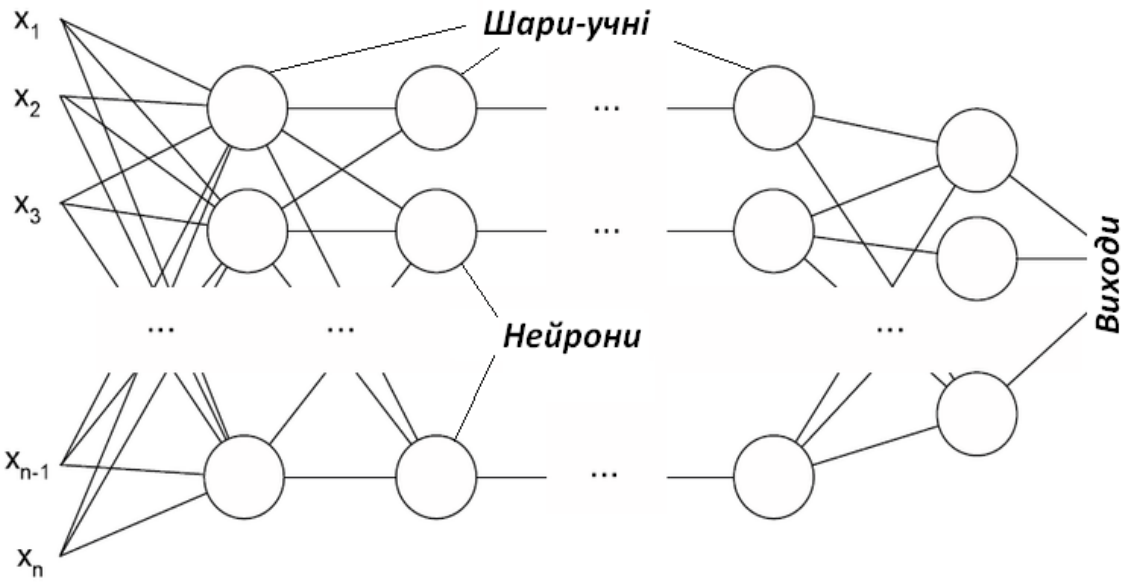

Рисунок 1.4 – *Багатошаровий персептрон*

В залежності від функцій, які виконують нейрони в мережі,згідно з [8], виділяють три види нейронів:

- вхідні нейрони (нейрони вхідного шару), на які подається вектор, що кодує вхідний вплив чи образ зовнішнього середовища; в них, зазвичай, не здійснюється обчислювальних процедур, а інформація передається з входу на вихід шляхом зміни їх активації;

- вихідні нейрони (нейрони вихідного шару), вихідні значення яких представляють виходи нейронної мережі;

- проміжні нейрони (нейрони прихованих шарів, або шарів-учнів), що складають основу нейронної мережі; в таких нейронах виконуються перетворення.

2.4 Навчання нейронних мереж

Здатність до навчання є фундаментальною властивістю мозку людини. В контексті штучних нейронних мереж процес навчання розглядається як налаштування архітектури мережі та ваг зв’язків для ефективного виконання спеціалізованої задачі. Здебільшого, нейронна мережа повинна налаштувати ваги зв’язків за наявною навчальною вибіркою. Функціонування мережі покращується по мірі ітеративного корегування вагових коефіцієнтів. Властивість мережі навчатися на прикладах робить її більш привабливою в порівнянні з системами, які слідують певній системі правил функціонування,

сформульованій експертами. Вище описану властивість до навчання нейронної мережі називають адаптивністю.

Отже навчання нейронної мережі полягає у знаходженні значень ваг (інколи - порогів), які б мінімізували помилку прогнозу, що видається мережею. По суті цей процес є підгонкою моделі, яка реалізується мережею, до наявних навчальних даних. Помилка для конкретної конфігурації мережі визначається шляхом прогону через мережу всіх наявних спостережень і порівняння вихідних значень з бажаними (цільовими). Всі такі різниці підсумовуються в так звану функцію помилок, значення якої і є помилкою мережі. В якості функції помилок найчастіше береться сума квадратів помилок, тобто коли всі помилки вихідних елементів для всіх спостережень зводяться в квадрат і потім додаються.

Одна з найбільш серйозних проблем прийнятого підходу полягає в тому, що таким чином ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати, - помилку, яку можна очікувати від мережі, коли їй будуть подаватися зовсім нові спостереження. Інакше кажучи, нейронна мережа повинна узагальнювати результат на нові спостереження. Насправді мережа навчається мінімізувати помилку на навчальній вибірці, і при відсутності ідеальної і нескінченно великої навчальної вибірки це зовсім не те ж саме, що мінімізувати "справжню" помилку в заздалегідь невідомої моделі явища. Найсильніше це розходження проявляється в проблемі перенавчання, або занадто близькою підгонки.

Описані недоліки та проблема з локальними мінімумами і вибором розміру мережі призводять до того, що при практичній роботі з нейронними мережами, як правило, доводиться експериментувати з великим числом різних мереж, часом навчаючи кожну з них по кілька разів (щоб не бути введеним в оману локальними мінімумами) і порівнюючи отримані результати. Головним показником якості результату є контрольна помилка. При цьому, відповідно до загальнонаукових принципів, згідно з якими при рівних моделях слід віддати перевагу більш простій, з двох мереж з приблизно рівними помилками контролю має сенс вибрати ту, яка менше.

Необхідність багаторазових експериментів веде до того, що контрольна вибірка починає відігравати ключову роль у виборі моделі, тобто стає частиною процесу навчання. Тим самим послаблюється її роль як незалежного критерію якості моделі – при великій кількості експериментів є ризик вибрати

невдалу мережу, що дає хороший результат на контрольній вибірці. Для того, щоб надати остаточній моделі належну надійність, часто (принаймні, коли обсяг навчальних даних це дозволяє) роблять так: резервують ще одну - тестову вибірку спостережень. Підсумкова модель тестується на даних з цієї вибірки, щоб переконатися, що результати, досягнуті на навчальній та контрольній вибірках реальні, а не є виключеннями процесу навчання. Зрозуміло, для того щоб добре грати свою роль, тестова вибірка повинна бути використана тільки один раз: якщо її використовувати повторно для коригування процесу навчання, то вона фактично перетвориться на контрольну вибірку.

На даний момент існує три парадигми навчання: «з вчителем», «без вчителя» (самонавчання) та змішане навчання. В першому випадку нейронна мережа має правильні відповіді (виходи мережі) на кожний приклад. Ваги налаштовуються так, щоб мережа надавала виходи якомога ближчі до відомих правильних виходів. Посилений варіант навчання з вчителем передбачає, що відома тільки критична оцінка правильності виходів нейронної мережі, але не самі правильні відповіді (виходи). Навчання без вчителя не потребує знання правильних відповідей на кожен приклад навчальної вибірки. В цьому випадку розкривається внутрішня структура даних або кореляції між образами в системі даних, що дозволяє розподілити образи по категоріям. Під час змішаного навчання частина ваг визначається навчанням з вчителем, в той час як інша отримується самонавчанням.

2.5 Алгоритм зворотного поширення помилки

Найвідоміший варіант алгоритму навчання нейронної мережі - так званий алгоритм зворотного поширення помилки. Існують також сучасні алгоритми, такі як метод сполучених градієнтів і метод Левенберга-Маркарова [10], які на багатьох задачах працюють значно швидше (іноді на порядок). Алгоритм зворотного поширення найбільш простий для розуміння, а в деяких випадках він має певні переваги.

Навчання алгоритмом зворотного поширення помилки припускає два проходи по всім шарам мережі: прямого і зворотного. При прямому проході вхідний вектор подається на вхідний шар нейронної мережі, після чого поширюється по мережі від шару до шару. В результаті генерується набір вихідних сигналів, який і є фактичною реакцією мережі на даний вхідний образ. Під час прямого проходу всі синаптичні ваги мережі фіксовані. Під час

зворотного проходу синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається від бажаного, в результаті чого формується сигнал помилки. Цей сигнал згодом поширюється по мережі в напрямку, зворотному напрямку синаптичних зв'язків. Синаптичні ваги налаштовуються з метою максимального наближення вихідного сигналу мережі до бажаного.

2.5.1 Постановка задачі

Розглянемо багатошарову мережу, в який кожен нейрон попереднього шару пов'язаний з усіма нейронами наступного шару (рис. 1.4). Позначимо . Нехай вихідний шар складається з m нейронів з виходами . Перед ним знаходиться прихований шар з H нейронів з функціями активації і виходами . Ваги синаптичних зв'язків між h-м нейроном прихованого шару і p-м нейроном вихідного шару будемо позначати через . Перед цим шаром може знаходитися або вхідний, або ще один прихований шар з виходами і синаптичними вагами . В загальному випадку кількість шарів може бути довільною. Вихідні значення мережі на об'єкті , розраховуються як суперпозиція:

$$a_p(x_i) = \sum_{h=0}^{H} w_{h,p} u^h(x_i), \qquad u^h(x_i) = \sigma_h\left(\sum_{j=0}^{J} w_{j,h} v^j(x_i) \right) \tag{2.2}$$

Метою метода зворотного поширення помилки, за методом найменших квадратів, є мінімізація цільової функції виду

$$Q(w) = \frac{1}{2} \sum_{p=1}^{m} \left(a^p(x_i) - y_i^p \right)^2 \tag{2.3}$$

2.5.2 Математична модель

Метод зворотного поширення помилки базується на градієнтному методі та полягає у налаштуванні ваг синаптичних зв’язків.

Надалі нам знадобляться часткові похідні функціонала Q по виходам нейронів. Запишемо спочатку для вихідного шару:

$$\frac{\partial Q(w)}{\partial a^p} = a^p(x_i) - y_i^p = \varepsilon_i^p \tag{2.4}$$

Тобто, часткова похідна Q по дорівнює величині помилки на об’єкті. Далі запишемо часткові похідні по виходам схованого шару:

$$\frac{\partial Q(w)}{\partial u^h} = \sum_{p=1}^{m} \left(a^p(x_i) - y_i^p \right) \cdot \sigma'_p \, w_{h,p} = \sum \varepsilon_i^p \cdot \sigma'_p \, w_{h,p} = \varepsilon_i^h \tag{2.5}$$

Цю величину, за аналогією з ε_i^h , будемо називати помилкою мережі на схованому шарі і позначати як ε_i^h . Через σ'_p позначимо похідну функції активації, розраховану за тим же значенням аргументу, що і (2.2). Якщо використовується сигмоїдальна функція активації, то для ефективного розрахунку похідної можна скористатися формулою $\sigma'_p = \sigma_p(1-\sigma_p)$.

Відмітимо, що ε_i^h розраховується через ε_i^p , якщо проходити мережу в зворотному напрямку, подаючи на виходи нейронів прихованого шару значення $\varepsilon_i^p \sigma'_p$ та отримуючи на вході результат ε_i^h . При цьому вхідний вектор скалярно множиться на вектор ваг $w_{h,p}$, що знаходяться «праворуч» від нейрона, як при прямому проходженні обчислень (рис 2.5).

Маючи часткові похідні по a^p і u_h , легко записати градієнт Q по вагам:

$$\frac{\partial Q(w)}{\partial w_{h,p}} = \frac{\partial Q(w)}{\partial a^p} \cdot \frac{\partial a^p}{\partial w_{h,p}} = \varepsilon_i^p \sigma'_p \, u^h, \; p = \overline{1,m}, h = \overline{0,H}; \tag{2.6}$$

$$\frac{\partial Q(w)}{\partial w_{j,h}} = \frac{\partial Q(w)}{\partial u^h} \cdot \frac{\partial u^h}{\partial w_{j,h}} = \varepsilon_i^h \sigma'_h \, v^j, h = \overline{1,H}, \; j = \overline{1,J}; \tag{2.7}$$

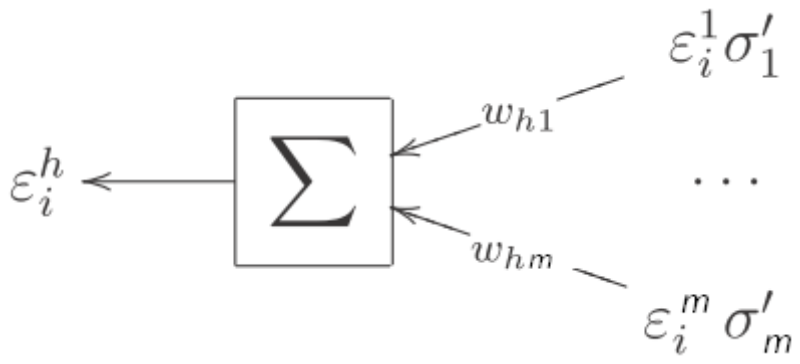

Рисунок 1.5 – *Схема зворотного розповсюдження*

Таким чином, отримано градієнт для двох шарів нейронної мережі. Якщо мережа має більше ніж два шари, то решта часткових похідних записується аналогічно – зворотнім ходом по мережі справа на ліво.

Запишемо алгоритм методу зворотного розповсюдження.

2.5.3 Алгоритм методу

Визначимо вхідні та вихідні дані до алгоритму метода зворотного розповсюдження помилки.

Вхідні дані:

- - навчальна вибірка,
- H – число нейронів в прихованому шарі;
- η – швидкість (темп) навчання;

Вихідні дані:

- синаптичні ваги

Алгоритм описаний в [9] зобразимо у вигляді блок-схеми (рис. 1.6).

Слід також зауважити, що в алгоритмі зворотного поширення обчислюється вектор градієнта поверхні помилок. Цей вектор вказує напрямок найкоротшого спуску по поверхні з даної точки, тому якщо ми "трохи" просунемося по ньому, помилка зменшиться. Певні труднощі тут представляє питання про те, яку треба брати довжину кроків.

При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути через рішення або (якщо поверхня помилок має особливо химерну форму) піти в неправильному напрямку. Класичним прикладом такого явища при навчанні нейронної мережі є ситуація, коли алгоритм дуже повільно просувається по вузькому яру і, навпаки, при маленькому кроці, ймовірно, буде схоплено вірний напрям, однак при цьому буде потрібно дуже багато ітерацій. На практиці величина кроку береться пропорційної крутизні схилу (так що алгоритм уповільнює хід поблизу мінімуму) з деякою константою η, яка називається швидкістю навчання. Ця константа може також залежати від часу, зменшуючись по мірі просування алгоритму.

Зазвичай цей алгоритм видозмінюється таким чином, щоб включати доданок імпульсу (або інерції). Цей член сприяє просуванню у фіксованому напрямку, тому якщо було зроблено кілька кроків в одному і тому ж напрямку, то алгоритм "збільшує швидкість", що (іноді) дозволяє уникнути локального мінімуму, а також швидше проходити плоскі ділянки.

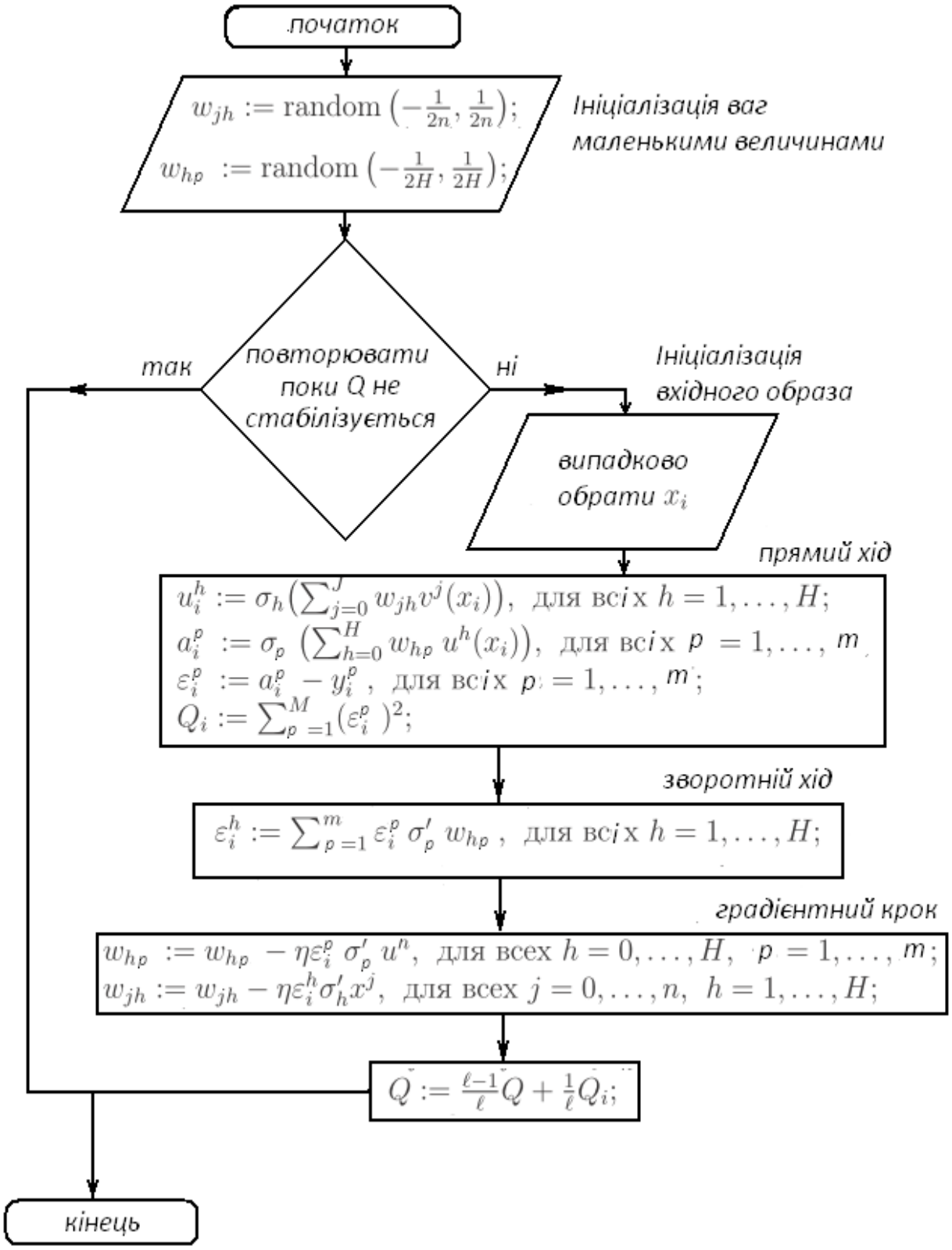

Рисунок 1.6 – *Алгоритм зворотного розповсюдження помилки*

Висновок до розділу 2

В даному розділі приведено необхідні теоретичні дані для побудови класифікатора за допомогою багатошарової нейронної мережі. Наведені основні принципи побудови та навчання нейронних мереж. Був розглянутий метод зворотного розповсюдження помилки та наведений алгоритм цього методу, який використовується в багатьох штучних нейронних мережах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Мак-Каллок У.С., Питтс В. Логическое исчисление идей, относящихся к нервной активности // В сб.: «Автоматы» под ред. К.Э. Шеннона и Дж. Маккарти. – М.: Изд-во иностр. лит., 1956. – с.363–384.

2. J. Hertz, A. Krogh, and R.G. Palmer, Introduction to the Theory of Neural Computation. – Addison-Wesley, Reading, Mass., 1991. – 111 p.

3. Хайкин С. Нейронные сети. Полный курс / Хайкин С. – М. : Вильямс, 2006. – 1104 с.

4. Каллан Р. Основные концепции нейронных сетей / Роберт Каллан – М. : Вильямс, 2001. – 287 с.

5. R.Rosenblatt, "Principles of Neurodynamics", Spartan Books, New York, 1962.

6. М. Минский Персептроны / М. Минский, С. Пейперт – М. : Мир, 1971. – 261 с.

7. Гуляев Ю.В. Нейрокомпьютеры в системах обработки сигналов. Кн.9. Коллективная монография / Подред. Ю.В. Гуляева и А.И. Галушкина. М.: Радиотехника, 2003. 224 с.: ил.

8. Круглов В.В. Искусственные нейронные сети. Теория и пратика / Круглов В.В. Борисов В.В. – 2-е изд. – М. : Горячая линия – Телеком, 2002. – 382 с.: ил.

9. Панкай Гупта, Ашиш Гоель, Джиммі Лін, Анееш Шарма, Донг Ван та Реза Босаг Заде WTF: Система, яка слідкує в Twitter , Матеріали 22-ї міжнародної конференції

10. Х. Чен, Л. Гоу, Х. Чжан, К. Джайлз Колабсер: пошукова система для виявлення співпраці , в рамках спільної конференції ACM / IEEE про цифрові бібліотеки (JCDL) 2011

11. Рубенс, Ніл; Елахі , Мехді; Сугіяма, Масаші; Каплан, Дайн (2016). "Активне навчання в системах рекомендацій" . У Річчі, Франческо; Рокач, Ліор; Шапіра, Брача (ред.). Посібник із систем рекомендацій (2 видання). Спрінгер США

12. Хілл, Вілл, Ларрі Стід, Марк Розенштейн та Джордж Фурнас. " Рекомендування та оцінка вибору у віртуальній спільноті використання ." У матеріалах конференції SIGCHI про людські фактори в обчислювальних системах, с. 194-201. ACM Press / Addison-Wesley Publishing Co., 1995