

Міністерство освіти і науки України  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

**СЕРЕДА Юрій Іванович**

**«Засоби збору та візуалізації даних в  
системах управління / Means of data collection  
and visualization in control systems»**

Студент групи КІм – 21  
СЕРЕДА Юрій Іванович

---

Науковий керівник  
д.т.н., професор Цмоць І.Г.

---

**Тернопіль – 2020**

## РЕЗЮМЕ

Кваліфікаційна робота на тему “ Засоби збору та візуалізації даних в системах управління енергоефективністю” зі спеціальності<sup>123</sup> «Комп’ютерна інженерія» освітнього ступеня «магістр» написана обсягом 83 сторінка і містить 12 ілюстрацій, 3 таблиць, 2 додатків та 52 джерел за переліком посилань.

Мета роботи – розроблення засобів збору та візуалізації енергетичних даних для системи управління енергоефективністю регіону.

Методи досліджень. Для розв’язання поставлених задач у кваліфікаційній роботі використано: методи: лінійної алгебри та аналітичної геометрії (для створення моделей опису даних електроспоживання); математичного аналізу (опрацювання даних); об’єктно-орієнтованого програмування(для проектування програмних засобів вед-інтерфейсу). Результати дослідження: система збору аналізу та візуалізації даних із електромереж.

Результати роботи можуть бути використані для покращення роботи систем електропостачання та промисловості.

Орієнтовні напрямки розвитку досліджень: розроблення спеціалізованих систем аналізу промислових даних регіону; розширення інформаційно-аналітичних систем шляхом створення розподілених баз даних для зберігання інформації, створення засобів мультиконтрольності.

**КЛЮЧОВІ СЛОВА:** ЗБІР, ОПРАЦЮВАННЯ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ ЕНЕРГОЕФЕКТИВНОСТІ; ПІДТРИМКА ПРИЙНЯТТЯ УПРАВЛІНСЬКИХ РІШЕНЬ; КОНЦЕПЦІЯ ІНТЕРНЕТУ РЕЧЕЙ, ЗАСОБИ ГЕОІНФОРМАЦІЙНОЇ СИСТЕМИ.

## RESUME

Qualification work on "Means of data collection and visualization in energy efficiency management systems" in the specialty<sup>123</sup> "Computer Engineering" educational degree "Master" is written in 83 pages and contains 12 illustrations, 3 tables, 2 appendices and 52 sources from the list of references.

The purpose of the work is to develop means for collecting and visualizing energy data for the energy efficiency management system of the region.

Research methods. To solve the tasks in the qualification work used: methods: linear algebra and analytical geometry (to create models for describing power consumption data); mathematical analysis (data processing); object-oriented programming (for designing Web-interface software). Research results: system for collecting analysis and visualization of data from power grids.

The results of the work can be used to improve the performance of power supply systems and industry.

Approximate directions of research development: development of specialized systems of analysis of industrial data of the region; expansion of information-analytical systems by creating distributed databases for information storage, creating means of multicontrol.

**KEY WORDS: COLLECTION, PROCESSING AND VISUALIZATION OF ENERGY EFFICIENCY DATA; SUPPORT FOR MANAGEMENT DECISIONS; THE CONCEPT OF THE INTERNET OF THINGS, MEANS OF GEOINFORMATION SYSTEM..**

## ЗМІСТ

Вступ.....	13
1. Аналіз апарату штучних нейронних мереж.....	18
1.1 Актуальність штучних нейронних мереж.....	18
1.2 Порівняння продуктивності людського мозку і комп'ютера.....	19
1.3 Властивості штучних нейронних мереж.....	20
1.4 Переваги штучних нейронних мереж.....	22
1.5 Компоненти штучного нейрона.....	23
1.6 Архітектури з'єднань штучних нейронів.....	27
1.7 Навчання штучної нейронної мережі.....	31
1.8 Використання штучних нейронних мереж.....	33
1.9 Висновки до розділу.....	37
2. Архітектура інформаційно-аналітичної системи управління енергоефективністю економіки регіону.....	38
2.1 Вибір принципів та формування вимог до компонентів комп'ютерних штучних нейронних мереж реального часу.....	38
2.2 Вибір елементної бази та варіантів побудови компонентів комп'ютерних штучних нейронних мереж реального часу.....	39
2.3 Система підтримки прийняття рішень для системи енергоуправління.....	42
2.4 Аналіз елементної бази для апаратних реалізацій нейромереж.....	45
2.5 Вибір засобів для візуалізації енергетичних даних.....	50
2.6 Висновки до розділу.....	55
3 Система підтримки прийняття рішень для контролю енергомережі.....	56
3.1 Реалізація апарату нейронних мереж.....	56
3.2 Розроблення засобів візуалізації просторових енергетичних даних.....	63
3.3 Висновки до розділу.....	73
Висновки.....	74
Список використаних джерел.....	75
Додаток А Текст програми опрацювання даних.....	81
Додаток Б Світлокопії публікації.....	84

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ГІС - геоінформаційні системи

POI - points of interest

ІАСУЕ - інформаційно-аналітичної системи управління енергоефективністю

KML - keyhole markup language

API - application programming interface

GPX - GPS eXchange

IAS - international accounting standards

GPS - global positioning system

ЕОМ - електронна обчислювальна машина

CSV- comma-separated values

НВІС - надвелика інтегральна схема

JSON - javascript object notation

ШНМ - штучна нейронна мережа

XML - extensible markup language

ООС - однорідних обчислювальних середовищ

ФОП – функціонально-орієнтований

БПП – буферна паралельна пам'ять

ІАД – інтелектуальний аналіз даних

OLAP - online analytical processing

ETL – extract, transform, and load

VME - versamodule eurocard

TCP/IP - transmission control protocol /internet protocol

HTTP - hyper text transfer protocol

NNP - net national product

БШП - багатошаровий перцептрон

ISA - industry standard architecture

AWS - amazon web services

MIPS - microprocessor without interlocked pipeline stages

GIPS - billion instructions per second

RBF - radially based functions

SAND - simple applicable neural device

DSP - digital signal processing

SHARC - super harvard architecture

SIMD - single instruction, multiple data

ММАС - мільйонів множень з накопиченням за секунду

ПЛІС - програмована логічна інтегральна схема

ПП – паралельна пам'ять

БПП – буферна паралельна пам'ять

## ВСТУП

Актуальність теми. Щоб підвищити енергоефективність в економічних показниках регіону використовують апарат аналітики значної кількості інформації, які базуються на інтелектуальних інформаційних, Web та телекомунікаційних технологіях. На основі масштабування модулів, доступності, сумісності та використання стандартних проектних рішень, буде розроблено концепт інформаційно-аналітичної системи управління енергоефективністю (ІАСУЕ) економіки регіону, що буде збирати інформацію, опр'юувати та візуалізовувати енергетичні данні, моделювати, прогнозувати процеси управління енергоефективністю та підтримку у прийнятті управлінських рішень. Потрібно структурувати інформацію і організувати її подачу єдиним простором з достовірністю, повнотою та оперативністю, яка потрібна для створення ефективних управлінських рішень. Сучасну концепцію інтернет речей прийнято, як можливе рішення для збору даних, основою яких є розподілені в просторі малі інтелектуальні сенсори, які пов'язані із центрами зберігання даних хмарних сервісів. Показано, що розроблення компонентів територіально орієнтованої системи для ІАСУЕ економіки доцільно застосувати з використанням сервісу віртуального зберігання даних Google Cloud та розгалуженого застосунку Google Maps API, що приведе до швидкого контактування компонентів, їх розширення та перетворення інформаційних функцій. Потрібно довести, що додаткове запровадження засобів програмування, таких як мови JavaScript з використанням Google Maps API може дозволити розробку потрібної нам системи для ІАСУЕ економіки регіону з різноманітних, не завжди прогнозованих вимог користувача системи у майбутньому. Дельтовидним буде забезпечити шляхи прийняття рішень ІАСУЕ економіки регіону орієнтовані на бази та сховища даних, певним чином прилаштованих та доступних засобів розробки ГІС для унаочнення та аналізу даних про енергоспоживання та енергоефективність, що дозволить логічно та ефективно приймати рішення. Метою ще показати візуалізовані дані результати

обробки у зрозумілому та користувацько дружньому вигляді з геолокаційним позиціонуванням об'єктів управління, що має дозволити потрібний рівень прийняття рішень.

По плану підвищення енергоефективності здійснюється завдяки використанню інформаційно-аналітичних додатків підтримки управління, як базис на інтелектуальній інформаційній, веб- та телекомунікаційній технологіях. Архітектура інформаційно-аналітичної системи (IAS) для управління енергоефективністю економіки регіону розроблена на основі правил модульності, відкритості, сумісності та використання набору базових проектних рішень. IAS забезпечує збір, обробку та унаочнення енергетичних даних, моделювання, прогностика енергоефективності та підтримка рішень управління енергоефективністю для регіональної економіки. Потрібно створити єдине інформаційне поле з надійною, повною та своєчасною інформацією, яка використовується для створення ефективних управлінських рішень. На основі концепції Internet of Things розроблені збирачі даних, що представляють собою просторово розподілені невеликі інтелектуальні датчики, пов'язані з хмарним сервером. Показано, що компоненти геоінформаційної системи для економіки регіону IACEA доцільно розробити за допомогою Cloud Cloud Services та спеціалізованого API Google Maps, що забезпечить оперативне створення, модифікацію та збільшення інформаційних можливостей. Стверджується, що додаткове залучення інструментів програмування, включаючи JavaScript, за допомогою API Google Maps надає можливість розробити геоінформаційну систему для IAS для підтримки управління енергоефективністю регіональної економіки з урахуванням додаткових конкретних майбутніх вимог цієї системи. Запропоновано створення IAS для підтримки управління енергоефективністю на базі даних та сховищ даних, спеціалізованих загальнодоступних інструментів ГІС унаочнення та аналізу даних про споживання енергії та енергоефективності, що забезпечить доцільність та ефективність створених рішень. Візуалізація даних про енергію та обробка результатів у найбільш



зрозумілій для людини формі з точним розташуванням засобів управління забезпечує ефективну підтримку управлінських рішень.

Постановка проблеми. Паливно-енергетичний комплекс є одним з основних фундаментів соціально-економічного розвитку України та її регіонів. Підвищення ефективності використання паливно-енергетичних ресурсів підвищує рівень соціально-економічного розвитку країни та регіону. Покращення енергетичних показників регіону забезпечиться використанням інформаційно-аналітичних засобів прийняття рішень, які базуються на інтелектуальних інформаційних, Web та телекомунікаційних технологіях. З використанням вказаних технологій пропонується створення системи управління енергоефективністю (ІАСУЕ) економіки регіону, яка повинна забезпечувати збирання, роботу та унаочнення енергетичних даних, моделювання, прогнозування процесів управління і, як результат – підтримку управлінських рішень направлених на підвищення енергоефективності економіки регіону. При управлінні енергоефективністю економіки області повинна використовуватися інтеграція даних підприємств транспорту, промисловості та комунальників регіону. Основними задачами, які розв'язуються ІАСУЕ регіону є:

1. збір та збереження даних від транспортників, та інших дотичних підприємств у режимі реального часу;
2. унаочнення та обробка енергетичних даних;
3. робота по інформуванню з підприємствами;
4. швидкий доступ до даних підприємств;
5. опрацювання даних в динаміці від підприємства регіону;
6. динаміка зміни ефективності регіону;
7. моделювання та прогнозування енергоефективності підприємств і економіки регіону;
8. оперативний контроль енергоспоживання;
9. оперативне виявлення втрат енергоносіїв;
10. підтримка прийняття рішень з підвищення енергоефективності.

Важливим інструментом підвищення енергоефективності економіки регіону

є використання ІАСУЕ, яка орієнтована на розвиток економіки регіону. Підвищення енергоефективності економіки регіону є комплексною проблемою, вирішення якої передбачає збалансоване і ощадливе споживання енергетичних ресурсів, формування та реалізацію управлінських рішень.

У зв'язку з цим особливої актуальності набуває проблема підвищення енергоефективності економіки регіону шляхом розроблення та використання засобів інформаційно-аналітичної підтримки управління енергоефективністю регіону.

Аналіз останніх досліджень та публікацій. За останні роки здійснено багато досліджень та публікацій, присвячених питанням розроблення та використання засобів інформаційно-аналітичної підтримки управління енергоефективністю регіону. Розглянемо найважливіші з цих публікацій.

У роботах [3-13, 19, 20] проаналізовано принципи побудови та архітектуру ІАСУЕ, основними з яких є модульність побудови, змінний склад обладнання, відкритість і сумісність. Недоліком існуючих інформаційно-аналітичних систем є неповнота автоматизації процесів управління, які передбачають збір та опрацювання даних.

Різноманітні задачі візуалізації детально розглянуто в публікаціях [6, 14, 15, 18, 21-26]. З аналізу вказаних публікацій випливає, що у більшості випадків зібрані енергетичні дані подаються у вигляді таблиць, які використовуються для подальшого опрацювання. Недоліком таких засобів візуалізації є те, що вони не відтворюють у максимально сприйнятливому для людини вигляді зібрані дані та результати їх опрацювання, які доцільно подавати в вигляді графіків, діаграм, картографічних даних з точною прив'язкою до місць розташування об'єктів управління.

З проведеного аналізу випливає, що для підтримки прийняття ефективних управлінських рішень, необхідно розробити систему, яка забезпечувала б збір, опрацювання та візуалізацію енергетичних даних і результатів їх обробки у максимально сприйнятному для людини вигляді.

Об'єкт дослідження - дані енергоефективності

Предмет дослідження - засоби збору та візуалізації даних.

Мета та задачі дослідження. Мета роботи – розроблення засобів збору та візуалізації енергетичних даних для системи управління енергоефективністю регіону. Для досягнення мети необхідно вирішити наступні задачі:

- проаналізувати методи та алгоритми збору даних;
- провести аналіз підходів до аналізу даних енергоспоживання;
- провести дослідження існуючих програмних систем синтезу, аналізу та обробки даних по енергоспоживанню;
- проаналізувати існуючі алгоритми перетворення створення нейронних мереж;
- розробити алгоритм збору, аналізу та візуалізації даних;
- реалізувати програмну систему яка буде виконувати дані алгоритми.

Наукова новизна одержаних результатів. Розроблено алгоритми збору, аналізу та візуалізації даних.

Практичне значення отриманих результатів. Спроектовано систему програмну систему реалізує алгоритми збору, аналізу та візуалізації даних.

Публікації та апробація випускної кваліфікаційної роботи. Отримані результати апробовані в межах III науково-практичної конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі» Західноукраїнського національного економічного університету та опубліковано дві тези доповіді по темі роботи [1,2].

# 1. АНАЛІЗ АПАРАТУ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

## 1.1 Актуальність штучних нейронних мереж

Дослідження по штучних нейронних мережах пов'язані з тим, що спосіб обробки інформації людським мозком принципово відрізняється від методів, що застосовуються звичайними цифровими комп'ютерами. Мозок є надзвичайно складним, нелінійним, паралельним комп'ютером. Вченими доведено [1], що мозок складається з величезної кількості структурних компонентів – нервових клітин, що називаються нейронами. Нейрон створює з'єднання з іншими нейронами, кількість яких може бути від 10 до 100 000 в точках сполучення. Сигнали, що розповсюджуються по нейронній мережі, керують активністю мозку впродовж короткого інтервалу, а також стають причиною довготривалих змін стану самих нейронів і їхніх з'єднань.

Мозок людини має здатність організувати роботу нейронів, так, щоб вони могли виконувати конкретні задачі (такі як розпізнавання образів, обробку сигналів органів відчуттів, моторні функції) у багато разів швидше, ніж можуть дозволити найбільш швидкі сучасні комп'ютери. Прикладом такої задачі обробки інформації може бути звичайний зір [1,2]. У функції зорової системи входить створення представлення довколишнього світу у такому вигляді, який забезпечує можливість взаємодіяти з цим світом. Більш точно, мозок послідовно виконує ряд задач розпізнавання (наприклад, розпізнавання знайомого обличчя в незнайомій обстановці). На це у нього йде близько 100-200 мілісекунд, в той же час як виконання аналогічних задач навіть меншої складності на комп'ютері може зайняти кілька днів.

Іншим прикладом може бути локатор летючої миші, що являє собою систему активної ехолокації. Крім надання інформації про відстань до потрібного об'єкту, цей локатор надає інформацію про відносну швидкість об'єкту, його розміри та розміри його окремих елементів, а також про азимут і висоту руху об'єкта. Для виділення цієї інформації з отриманого сигналу маленький мозок

летючої миші здійснює складні нейронні обчислення. Ехолокація летючої миші за своїми характеристиками якості і швидкодії переважає найскладніші прилади, що створені інженерами.

Розробки штучних нейронних мереж почалися на початку ХХ століття, але тільки в 90-ті роки, коли обчислювальні системи стали достатньо потужними, нейронні мережі отримали широке розповсюдження [1]. Створення нейронних мереж було викликано спробами зрозуміти принципи роботи людського мозку і, без сумніву, це буде впливати і на подальший їх розвиток. У порівнянні з людським мозком нейронна мережа сьогодні є досить спрощеною моделлю, але, незважаючи на це, досить успішно використовується для вирішення найрізноманітніших задач. Хоча вирішення на основі нейронних мереж можуть виглядати і вести себе, як звичайне програмне забезпечення, вони відрізняються в принципі, оскільки більшість реалізацій на основі нейронних мереж "навчається", а не "програмується": мережа вчиться виконувати задачу, а не програмується безпосередньо [3].

## 1.2 Порівняння продуктивності людського мозку і комп'ютера

Мозок і цифровий комп'ютер виконують зовсім різні задачі і мають різні властивості. У типовому мозку людини є в 1000 разів більше нейронів, ніж логічних елементів у процесорі типового комп'ютера високого класу. У відповідності із законом Мура та з врахуванням того, що за деякими розрахунками кількість нейронів у мозку повинні подвоюватися приблизно через кожні 2-4 мільйони років, можна зробити прогноз, що кількість логічних елементів у процесорі стане рівною кількості нейронів у мозку приблизно до 2020 року [3]. Безумовно, ці прогнози мало про що говорять, крім того, ця різниця у відношенні кількості елементів є незначною у порівнянні з різницею часу переключання і ступенем розпаралелювання. Мікросхеми комп'ютера здатні виконати окрему команду менше ніж за наносекунду, тоді як нейрони діють в

мільйони разів повільніше. Але мозок сторицею компенсує цей свій недолік, оскільки всі його нейрони діють одночасно, тоді як більшість сучасних комп'ютерів має тільки один процесор (але з кількома ядрами) або невелику їх кількість. Таким чином, навіть незважаючи на те, що комп'ютер має перевагу більш ніж у мільйон разів у фізичній швидкості переключання, виявляється, що мозок у порівнянні з ним виконує всі свої дії приблизно в 100 000 разів швидше.

### 1.3 Властивості штучних нейронних мереж

Штучні нейронні мережі індуковані біологією, оскільки вони складаються з елементів, функціональні можливості яких аналогічні більшості елементарних функцій біологічного нейрона. Ці елементи потім організуються за способом, який відповідає анатомії мозку. Навіть при такій поверхневій подібності, штучні нейронні мережі демонструють дивовижну кількість властивостей, що притаманні мозку. Наприклад, вони навчаються на основі досвіду, узагальнюють попередні прецеденти на нові випадки і видобувають суттєві властивості із вхідної інформації, що містить лишні дані.

Незважаючи на таку функціональну подібність, навіть найоптимістичніший захисник не припустить, що в ближчому майбутньому штучні нейронні мережі будуть дублювати функції людського мозку. Реальний «інтелект», що демонструється найскладнішими нейронними мережами, знаходиться нижче рівня дощового черв'яка, і ентузіазм має бути поміркованим у відповідності до сучасних реалій. Але, тим не менше, було би неправильно ігнорувати дивовижну подібність у функціонуванні деяких нейронних мереж з людським мозком. Ці можливості, як би вони не були обмежені сьогодні, приводять до думки, що глибоке проникнення у людський інтелект, а також велика кількість революційних програм, можуть бути не за горами. Отже, розглянемо деякі властивості нейронних мереж [4].

1. Навчання - штучні нейронні мережі можуть міняти свою поведінку у залежності від зовнішнього середовища. Цей фактор у більшій степені, ніж будь-який інший, відповідає за ту цікавість, який вони викликають. Після представлення вхідних сигналів (можливо, разом з потрібними виходами) вони самоналаштовуються, щоб забезпечити потрібну реакцію. Було розроблено багато навчальних алгоритмів, кожний зі своїми сильними та слабкими сторонами. Все ще існують проблеми відносно того, чому мережі можна навчитися і як навчання повинне відбуватися.

2. Узагальнення - відгук мережі після навчання може бути до деякої степені нечутливим до невеликих змін вхідних сигналів. Ця внутрішньо властива здатність бачити образ крізь шум і спотворення життєво необхідна для розпізнавання образів у реальному світі. Вона дозволяє подолати вимогу строгої точності, що висувається звичайним комп'ютером, і відкриває шлях до системи, яка може мати справу з тим недосконалим світом, в якому ми живемо. Важливо зауважити, що штучна нейронна мережа робить узагальнення автоматично завдяки своїй структурі, а не за допомогою використання «людського інтелекту» у формі спеціально написаних комп'ютерних програм.

3. Абстрагування - деякі зі штучних нейронних мереж мають здатність видобувати сутність із вхідних сигналів. Наприклад, мережа може бути навчена на послідовність викривлень версії букви «А». Після відповідного навчання представлення такого викривленого прикладу призведе до того, що мережа видасть букву ідеальної форми (в даному випадку букву «А»). В деякому сенсі вона навчиться породжувати те, що ніколи не бачила. Здатність видобувати ідеальні прототипи є у людей досить цінною якістю.

4. Застосовність - штучні нейронні мережі не є панацеєю. Вони, очевидно, не підходять для виконання таких задач, як нарахування заробітної плати, але вони незамінні у великому класі інших задач, з якими погано або взагалі не справляються звичайні обчислювальні системи.

## 1.4 Переваги штучних нейронних мереж

Очевидно, що свою силу нейронні мережі черпають, по-перше, з розпаралелювання обробки інформації і, по-друге, із здатності самонавчатися, тобто, створювати узагальнення [5]. Під терміном узагальнення розуміють здатність отримувати обґрунтований результат на основі даних, які не зустрічалися у процесі навчання. Ці властивості дозволяють нейронним мережам вирішувати складні (масштабні) задачі, які на сьогодні вважаються важко вирішуваними. Але на практиці при автономній роботі нейронні мережі не можуть забезпечити готові рішення. Їх потрібно інтегрувати у складні системи. Зокрема, комплексну задачу можна розбити на послідовність відносно простих, частина з яких може вирішуватися нейронними мережами.

Далі наведені деякі переваги штучних нейронних мереж над традиційними обчислювальними системами [1-4].

1. Вирішення задач при невідомих закономірностях. Використовуючи навчання на великій кількості прикладів, нейронна мережа здатна вирішувати задачі, у яких невідомі закономірності розвитку ситуації і залежності між вхідними і вихідними даними. Традиційні математичні методи і експертні системи у таких випадках безпорадні.

2. Стійкість до шуму у вхідних даних. Можливість роботи при наявності великої кількості неінформативних, шумових вхідних сигналів. Немає необхідності робити їхній попередній відсів, нейронна мережа сама визначить їхню малу придатність для вирішення задачі і відкине їх.

3. Адаптування до змін навколишнього середовища. Нейронні мережі мають здатність адаптуватися до змін навколишнього середовища. Зокрема, нейронні мережі, навчені діяти у певному середовищі, можуть бути легко перенавчені для роботи в умовах незначних коливань параметрів середовища. Більш того, для роботи в нестаціонарному середовищі (де статистика змінюється з часом) можуть бути створені нейронні мережі, що перенавчаються в реальному



часі. Чим вищі адаптивні здатності системи, тим більш стійкою буде робота в нестационарному середовищі. При цьому потрібно зауважити, що адаптивність не завжди веде до стійкості; деколи вона призводить до зовсім протилежного результату. Наприклад, адаптивна система з параметрами, що швидко змінюються в часі, може також швидко реагувати і на сторонні збудження, що викличе втрату продуктивності. Для того щоб використовувати всі переваги адаптивності, основні параметри системи повинні бути достатньо стабільними, щоб можна було не враховувати зовнішні завади, і достатньо гнучкими, щоб забезпечити реакцію на суттєві зміни середовища.

4. Потенційно надвелика швидкодія. Нейронні мережі мають потенційно надвелику швидкодію за рахунок використання масового паралелізму обробки інформації.

5. Завадостійкість при апаратній реалізації. Нейронні мережі потенційно завадостійкі. Це означає, що при неблагополучних умовах їхня продуктивність падає несуттєво. Наприклад, якщо пошкоджений якийсь нейрон або його зв'язки, видобування збереженої інформації ускладнюється. Але, зважаючи на розподілений характер збереження інформації в нейронній мережі, можна стверджувати, що тільки серйозне пошкодження структури нейронної мережі суттєво вплине на її працездатність. Тому зменшення якості роботи нейронної мережі відбувається повільно.

## 1.5 Компоненти штучного нейрона

Незалежно від розташування та функціонального призначення, всі штучні нейронні елементи мають спільні компоненти [6,7]. Розглянемо сім основних компонентів штучного нейрона.

### Компонент 1. Вагові коефіцієнти

При функціонуванні нейрон одночасно отримує багато вхідних сигналів. Кожен вхід має свою власну синаптичну вагу, яка надає входу вплив, необхідний для функції суматора елемента обробки. Ваги є мірою сили вхідних зв'язків і моделюють різноманітні синаптичні сили біологічних нейронів. Ваги суттєвого входу підсилюються і, навпаки, вага несуттєвого входу примусово зменшується, що визначає інтенсивність вхідного сигналу. Ваги можуть змінюватись відповідно до навчальних прикладів, топології мережі та навчальних правил.

## Компонент 2. Функція суматора

Першим кроком дії нейрону є обчислення зваженої суми всіх входів. Математично, вхідні сигнали та відповідні їм ваги представлені векторами  $(x_1, x_2 \dots x_n)$  та  $(w_1, w_2 \dots w_n)$ . Добуток цих векторів є загальним вхідним сигналом. Спрощеною функцією суматора є множення кожного компонента вектора  $x$  на відповідний компонент вектора  $w$ : вхід 1 =  $x_1 * w_1$ , вхід 2 =  $x_2 * w_2$ , і знаходження суми всіх добутків: вхід1 + вхід2 + ... + вхід n. Результатом є єдине число, а не багатоелементний вектор.

Функція суматора може бути складнішою, наприклад, вибір мінімуму, максимуму, середнього арифметичного, добутку або виконувати інший нормалізуючий алгоритм. Вхідні сигнали та вагові коефіцієнти можуть комбінуватись багатьма способами перед надходженням до передатної функції. Особливі алгоритми для комбінування входів нейронів визначаються обраними мережною архітектурою та парадигмою.

В деяких нейромережах функції суматора виконують додаткову обробку, так звану функцію активації, яка зміщує вихід функції суматора відносно часу. На жаль, функції активації на теперішній час обмежено досліджені і більшість сучасних нейронних реалізацій використовують функцію активації "тотожності", яка еквівалентна її відсутності. Цю функцію доцільніше використовувати як компонент мережі в цілому, ніж як компонент окремого нейрона.

## Компонент 3. Передатна функція

Результат функції суматора є зваженою сумою вхідних сигналів, що перетворюється у вихідний сигнал через алгоритмічний процес відомий під назвою передатна функція. У передатній функції для визначення виходу нейрона загальна сума порівнюється з деяким порогом. Якщо сума є більшою за значення порога, елемент обробки генерує сигнал, в іншому випадку сигнал не генерується або генерується гальмуючий сигнал.

Переважно застосовують нелінійну передатну функцію, оскільки лінійні (прямолінійні) функції обмежені і вихід є просто пропорційним до входу. Застосування лінійних передатних функцій було проблемою у ранніх моделях мереж, і їх обмеженість та недоцільність була доведена в книзі Мінські та Пейперта "Перцептрони".

На рисунку 1.1 зображені типові передатні функції.

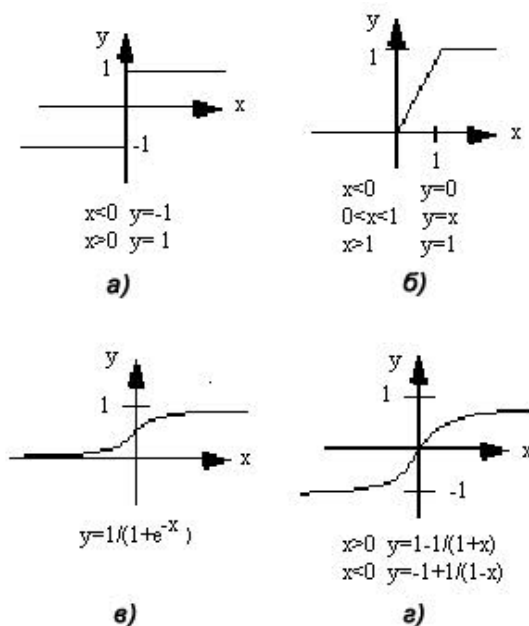


Рисунок 1.1 Типові передатні функції: жорстка порогова функція (а), лінійна з насиченням (б), сигмоїда (в), гіперболічний тангенс (г)

Для простої передатної функції нейромережа може видавати 0 та 1, 1 та -1 або інші числові комбінації. Передатна функція в таких випадках є "жорстким обмежувачем" або пороговою функцією (рисунок 1.1).

Інший тип передатної функції – лінійна з насиченням – віддзеркалює вхід всередині заданого діапазону і діє як жорсткий обмежувач за межами цього діапазону. Це лінійна функція, яка відсікається до мінімальних та максимальних значень, роблячи її нелінійною (рисунок 1.1б).

Наступним вибором є сигмоїда або S-подібна крива, яка наближує мінімальне та максимальне значення у асимптотах і називається сигмоїдою (рисунок 1.1в), коли її діапазон  $[0, 1]$ , або гіперболічним тангенсом (рисунок 1.1г), при діапазоні  $[-1, 1]$ . Важливою рисою цих кривих є неперервність функцій та їх похідних. Застосування сигмоїдних функцій надає добрі результати і має широке застосування.

Зрештою, для різних нейромереж можуть вибиратись інші передатні функції.

Перед надходженням до передатної функції до вхідного сигналу деколи додають однорідно розподілений випадковий шум, джерело та кількість якого визначається режимом навчання. В літературі цей шум згадується як "температура" штучних нейронів, яка надає математичній моделі елемент реальності.

#### Компонент 4. Масштабування

Після передатної функції вихідний сигнал проходить додаткову обробку масштабування, тобто результат передатної функції множиться на масштабуючий коефіцієнт і додається зміщення.

#### Компонент 5. Вихідна функція (змагання)

По аналогії з біологічним нейроном, кожний штучний нейрон має один вихідний сигнал, який передається до сотень інших нейронів. Переважно, вихід прямо пропорційний результату передатної функції. В деяких мережевих архітектурах результати передатної функції змінюються для створення змагання між сусідніми нейронами. Нейронам дозволяється змагатися між собою, блокуючи дії нейронів, що мають слабкий сигнал. Змагання (конкуренція) може відбуватись між нейронами, які знаходяться на одному або різних прошарках. По-

перше, конкуренція визначає, який штучний нейрон буде активним і забезпечить вихідний сигнал. По-друге, конкуруючі виходи допомагають визначити, який нейрон візьме участь у процесі навчання.

#### Компонент 6. Функція похибки та поширюване назад значення

У більшості мереж, що застосовують контрольоване навчання, обчислюється різниця між спродукованим та бажаним виходом. Похибка відхилення (біжуча похибка) перетворюється функцією похибки відповідно до заданої мережної архітектури. У базових архітектурах похибка відхилення використовується безпосередньо, в деяких парадигмах використовується квадрат або куб похибки зі збереженням знаку.

Після проходження всіх прошарків біжуча похибка поширюється назад до попереднього прошарку і може бути безпосередньо похибкою або похибкою, масштабованою певним чином залежно від типу мережі (наприклад, похідною від передатної функції). Це значення, що поширюється назад, враховується в наступному циклі навчання.

#### Компонент 7. Функція навчання

Метою функції навчання є налаштування змінних ваг з'єднань на входах кожного елемента обробки відповідно до певного алгоритму навчання для досягнення бажаного результату. Існує два типи навчання: контрольоване та неконтрольоване. Контрольоване навчання вимагає навчальної множини даних або спостерігача, що ранжує ефективність результатів мережі. У випадку неконтрольованого навчання система самоорганізовується за внутрішнім критерієм, закладеним в алгоритм навчання.

### 1.6 Архітектури з'єднань штучних нейронів

Об'єднуючись у мережі, нейрони утворюють системи обробки інформації, які забезпечують ефективну адаптацію моделі до постійних змін з боку

зовнішнього середовища. В процесі функціонування мережі відбувається перетворення вхідного вектора сигналів у вихідний [2]. Конкретний вид перетворення визначається як архітектурою нейромережі, так і характеристиками нейронних елементів, засобами керування та синхронізації інформаційних потоків між нейронами. Важливим фактором ефективності мережі є встановлення оптимальної кількості нейронів та типів зв'язків між ними.

При описі нейромереж використовують кілька усталених термінів, які в різних джерелах можуть мати різне трактування, зокрема:

1. структура нейромережі – спосіб зв'язків нейронів у нейромережі;
2. архітектура нейромережі – структура нейромережі та типи нейронів;
3. парадигма нейромережі – спосіб навчання та використання;

На основі однієї архітектури можуть бути реалізовані різні парадигми нейромережі і навпаки.

Серед відомих архітектурних рішень виділяють групу слабозв'язаних нейронних мереж, у випадку, коли кожний нейрон мережі зв'язаний лише із сусідніми.

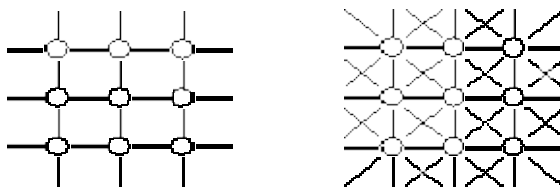


Рисунок 1.2 Слабозв'язані нейромережі

Навпаки, якщо входи кожного нейрона зв'язані з виходами усіх решта нейронів, то мова йде про повнозв'язані нейромережі.

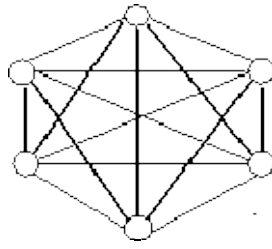


Рисунок 1.3 Повнозв'язні нейромережі

Зрозуміло, що такий поділ носить дещо теоретичний характер. Аналізуючи найбільш відомі на даний час розробки нейромереж [1-9], потрібно зазначити, що найпоширенішим варіантом архітектури є багатошарові мережі. Нейрони в даному випадку об'єднуються у прошарки з єдиним вектором сигналів входів. Зовнішній вхідний вектор подається на вхідний прошарок нейронної мережі (рецептори). Виходами нейронної мережі є вихідні сигнали останнього прошарку (ефектори). Окрім вхідного та вихідного прошарків, нейромережа має один або декілька прихованих прошарків нейронів, які не мають контактів із зовнішнім середовищем.

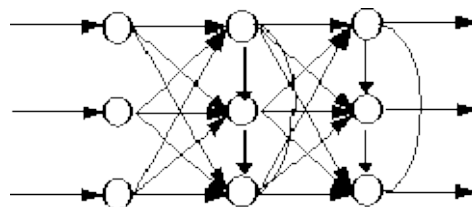


Рисунок 1.4 Багатошаровий тип з'єднання нейронів

1. зв'язки між нейронами різних прошарків називають проєктивними;
2. зв'язки, скеровані від вхідних прошарків до вихідних називаються аферентними;
3. зв'язки, скеровані від вихідних прошарків до вхідних називаються еферентними.
4. зв'язки між нейронами одного прошарку відносять до бічних (латеральних).

Фактично, за архітектурою зв'язків більшість відомих нейромереж, що знайшли практичне застосування, можна згрупувати у два великих класи [8]:

1. Мережі прямого поширення (з односкерованими послідовними зв'язками).

2. Мережі зворотного поширення (з рекурентними зв'язками).

Найбільш типовими мережами прямого поширення є перцептрон, мережа Back Propagation, мережа зустрічного поширення, карта Кохонена. Типовими рекурентними мережами є мережа Хопфілда, мережа адаптивної резонансної теорії, двоскерована асоціативна пам'ять.

У таблиці 1.1 подано назви найбільш типових архітектур мереж, що у свою чергу мають багато модифікацій та можуть бути складниками у інших мережах [9].

Таблиця 1.1 Найбільш відомі архітектури нейронних мереж

Мережі прямого поширення	Рекурентні мережі
перцептрони мережа Back Propagation мережа зустрічного поширення карта Кохонена	мережа Хопфілда мережа адаптивної резонансної теорії двоскерована асоціативна пам'ять

Мережі прямого поширення відносять до статичних, оскільки на задані входи нейронів надходить не залежний від попереднього стану мережі вектор вхідних сигналів.

Рекурентні мережі вважаються динамічними, тому що завдяки зворотнім зв'язкам (петлям) входи нейронів модифікуються в часі, що призводить до зміни станів мережі.

Оригінальність нейромереж, як аналога біологічного мозку, полягає у здатності до навчання за прикладами, що складають навчальну множину. Процес навчання нейромереж розглядається як налаштування архітектури та вагових



коефіцієнтів синаптичних зв'язків відповідно до даних навчальної множини так, щоб ефективно вирішити поставлену задачу.

## 1.7 Навчання штучної нейронної мережі

Контрольоване навчання. Величезна більшість рішень отримана від нейромереж з контрольованим навчанням, де біжучий вихід постійно порівнюється з бажаним виходом. Ваги на початку встановлюються випадково, але під час наступних ітерацій коректуються для досягнення близької відповідності між бажаним та біжучим виходом. Створені методи навчання спрямовані на мінімізацію біжучих похибок всіх елементів обробки, яка створюється за певний час неперервною зміною синаптичних ваг до досягнення прийнятної точності мережі [1, 6-8].

Перед використанням нейромережа з контрольованим навчанням повинна бути навченою. Фаза навчання може тривати багато часу, зокрема, у прототипах систем з невідповідною процесорною потужністю навчання може займати декілька годин. Навчання вважається закінченим при досягненні нейромережею визначеного користувачем рівня ефективності. Цей рівень означає, що мережа досягла бажаної статистичної точності, оскільки вона видає бажані виходи для заданої послідовності входів. Після навчання ваги з'єднань фіксуються для подальшого застосування. Деякі типи мереж дозволяють під час використання неперервне навчання, з набагато повільнішою оцінкою навчання, що допомагає мережі адаптуватись до умов, що повільно змінюються.

Навчальні множини повинні бути досить великими, щоб містити всю необхідну інформацію для виявлення важливих особливостей і зв'язків. Але і навчальні приклади повинні містити широке різноманіття даних. Якщо мережа

навчається лише для одного прикладу, ваги старанно встановлені для цього прикладу, радикально змінюються у навчанні для наступного прикладу. Попередні приклади при навчанні наступних просто забуваються. В результаті система повинна навчатись всьому разом, знаходячи найкращі вагові коефіцієнти для загальної множини прикладів. Наприклад, у навчанні системи розпізнавання піксельних образів для десяти цифр, які представлені двадцятьма прикладами кожної цифри, всі приклади цифри "сім" не доцільно представляти послідовно. Краще надати мережі спочатку один тип представлення всіх цифр, потім другий тип і так далі.

Головним для успішної роботи мережі є представлення і кодування вхідних і вихідних даних. Штучні мережі працюють лише з числовими вхідними даними, отже, необроблені дані, що надходять із зовнішнього середовища, повинні перетворюватись. Додатково необхідне масштабування, тобто нормалізація даних відповідно до діапазону всіх значень. Нормалізація виконується шляхом ділення кожної компоненти вхідного вектора на довжину вектора, що перетворює вхідний вектор в одиничний. Попередня обробка зовнішніх даних, отриманих за допомогою сенсорів, у машинний формат спільна для стандартних комп'ютерів і є легко доступною.

Якщо після контрольованого навчання нейромережа ефективно опрацьовує дані навчальної множини, важливим стає її ефективність при роботі з даними, які не використовувались для навчання. У випадку отримання незадовільних результатів для тестової множини, навчання продовжується. Тестування використовується для забезпечення запам'ятовування не лише даних заданої навчальної множини, але і створення загальних образів, що можуть міститися у даних.

Неконтрольоване навчання. Неконтрольоване навчання може бути великим надбанням у майбутньому. Суть його полягає у тому, що комп'ютери можуть самонавчатись у справжньому роботизованому сенсі [1, 5]. На даний час неконтрольоване навчання використовується в мережах, відомих як самоорганізовані карти (self organizing maps). Вони використовуються в досить

обмеженому користуванні, але доводять перспективність самоконтрольованого навчання. Мережі не використовують зовнішніх впливів для коректування своїх ваг і внутрішньо контролюють свою ефективність, шукаючи регулярність або тенденції у вхідних сигналах та роблять адаптацію згідно до навчальної функції. Навіть без повідомлення правильності чи неправильності дій, мережа повинна мати інформацію відносно власної організації, яка закладена у топологію мережі та навчальні правила.

Алгоритм неконтрольованого навчання скерований на знаходження близькості між групами нейронів, які працюють разом [4]. Якщо зовнішній сигнал активує будь-який вузол в групі нейронів, дія всієї групи в цілому збільшується. Аналогічно, якщо зовнішній сигнал в групі зменшується, це призводить до гальмуючого ефекту на всю групу.

Конкуренція між нейронами формує основу для навчання. Навчання конкуруючих нейронів підсилює відгуки певних груп на певні сигнали. Це пов'язує групи між собою та відгуком. При конкуренції змінюються ваги лише нейрона-переможця.

Оцінки навчання. Оцінка ефективності навчання нейромережі залежить від декількох керованих факторів [3, 9]. Теорія навчання розглядає три фундаментальні властивості, пов'язані з навчанням: ємність, складність зразків і обчислювальна складність. Під ємністю розуміють, скільки зразків може запам'ятати мережа, і які межі прийняття рішень можуть бути на ній сформовані. Складність зразків визначає число навчальних прикладів, необхідних для досягнення здатності мережі до узагальнення. Обчислювальна складність напряму пов'язана з потужністю процесора ЕОМ.

## 1.8 Використання штучних нейронних мереж

На сучасному етапі розвитку штучні нейронні мережі найчастіше використовуються для завдань розпізнавання і класифікації, прийняття рішень та

управління, кластеризації, прогнозування та апроксимації, і, також, стиснення даних та створення асоціативної пам'яті. Далі наведено детальне описання кожної з наведених задач [1-9].

Розпізнавання і класифікація. Образами можуть виступати різні за своєю природою об'єкти: символи тексту, зображення, зразки звуків і т.д. При навчанні мережі пропонуються різні зразки образів із вказанням того, до якого класу вони відносяться. Зразок, зазвичай, представляється як вектор значень ознак. При цьому сукупність всіх ознак повинна однозначно визначати клас, до якого відноситься образ. У випадку, якщо признаков недостатньо, мережа може співвіднести один і той самий зразок з кількома класами, що є неправильно. Після завершення навчання мережі їй можна представити невідомі раніше зразки і отримати відповідь про приналежність до певного класу.

Топологія такої мережі характеризується тим, що кількість нейронів у вихідному шарі, зазвичай, рівна кількості класів, що визначаються. При цьому встановлюється відповідність між виходом нейронної мережі і класом, який він представляє. Коли мережі подати певний образ, на одному з її виходів повинен з'явитися признак того, що образ належить цьому класу. У той же час на інших виходах має бути признак того, що образ даному класу не належить. Якщо на двох або більше виходах є признак належності до класу, вважається що мережа "не впевнена" у своїй відповіді.

Прийняття рішень і управління. Ця задача близька до задачі класифікації. Класифікації підлягають ситуації, характеристики яких поступають на вхід нейронної мережі. На виході мережі при цьому має з'явитися признак рішення, яке вона прийняла. При цьому в якості вхідних сигналів використовуються різні критерії описання стану системи, що керується.

Кластеризація. Під кластеризацією розуміємо розбиття множини вхідних сигналів на класи, при тому, що ні кількість, ні признаки наперед не відомі. Після навчання така мережа здатна визначати, до якого класу відноситься вхідний сигнал. Мережа також може сигналізувати про те, що вхідний сигнал не відноситься до жодного з виділених класів – це є ознакою нових, відсутніх у

навчальній вибірці, даних. Отже, така мережа може виявляти нові, невідомі раніше класи сигналів. Відповідність між класами, виділеними мережею, і класами, що існують у предметній області, задається людиною. Кластеризацію здійснюють, наприклад, нейронні мережі Кохонена.

Прогнозування та апроксимація. Здатність нейронної мережі до прогнозування напряму впливає із її здатності до узагальнення і виділення прихованих залежностей між вхідними і вихідними даними. Після навчання мережа здатна передбачати майбутні значення деякої послідовності на основі деяких попередніх значень і/або якихось існуючих на даний час факторів. Потрібно зауважити, що прогнозування можливе тільки тоді, коли попередні зміни дійсно певним чином визначають майбутнє. наприклад, прогнозування котувань акцій на основі котувань за попередній тиждень може виявитися успішним (а може і не виявитися), тоді як прогнозування результатів завтрашньої лотереї на основі даних за останні 50 років точно не дасть жодних результатів.

Стиснення даних та асоціативна пам'ять. Здатність нейромереж до виявлення взаємозв'язків між різними параметрами дає можливість виразити дані великої розмірності більш компактно, якщо дані тісно пов'язані одні з одними. Зворотній процес – відновлення вихідного набору даних з частини інформації – називається (авто)асоціативною пам'яттю. Асоціативна пам'ять дозволяє також відновлювати вихідний сигнал/образ із зашумлених/пошкоджених вхідних даних. Вирішення задачі гетероасоціативної пам'яті дозволяє реалізувати пам'ять, що адресується за вмістом.

В кожній предметній області, при ближчому розгляді, можна знайти постановки задач для нейронних мереж. Далі наведено список окремих областей, де вирішення такого роду задач мають практичне значення вже зараз [3-5].

Економіка та бізнес – прогнозування часових послідовностей (курсів валют, об'ємів продаж), автоматичний трейдинг, оцінка ризиків не повернення кредитів, передбачення банкрутств, оцінка вартості нерухомості, виявлення переоцінених і недооцінених компаній, автоматичне рейтингування, оптимізація товарних і

грошових потоків, автоматичне зчитування і розпізнавання чеків і документів, безпека транзакцій по пластикових картах.

Медицина – встановлення діагнозу, обробка медичних зображень, моніторинг стану пацієнта, аналіз ефективності лікування, очищення показників приладів від шумів.

Авіоніка – автопілоти, що навчаються, розпізнавання сигналів радарів, адаптивне пілотування сильно пошкодженого літака, безпілотні літальні апарати.

Зв'язок – стиснення відеоінформації, швидке кодування-декодування, оптимізація стільникових мереж і схем маршрутизації пакетів.

Інтернет – асоціативний пошук інформації, електронні секретарі та агенти користувача в Інтернет, фільтрація інформації, блокування спаму, автоматична рубрикація новинних стрічок, адресна реклама і маркетинг для електронної торгівлі.

Автоматизація виробництва – оптимізація режимів виробничого процесу, контроль якості продукції, моніторинг і візуалізація багатомірної диспетчерської інформації, попередження аварійних ситуацій, робототехніка.

Політологічні та соціологічні технології – передбачення результатів виборів, аналіз опитувань, передбачення динаміки рейтингів, виявлення вагомих факторів, кластеризація електорату, дослідження та візуалізація соціальної динаміки населення.

Безпека та охоронні системи – розпізнавання облич, ідентифікація особистості за відбитками пальців, голосом, підписом, розпізнавання автомобільних номерів, аналіз аерокосмічних фотографій, моніторинг інформаційних потоків у комп'ютерній мережі і виявлення вторгнень, виявлення підробок.

Введення та обробка інформації – розпізнавання рукописних текстів, поштових, платіжних, фінансових і бухгалтерських документів.

Геологорозвідка – аналіз сейсмічних даних, асоціативні методики пошуку корисних копалин, оцінка ресурсів родовищ.

### 1.9 Висновки до розділу

Найкращим підходом для реалізації поставлених завдань є використання нейронних мереж як апарат збору та обробки даних та комбіновані пристрої обробки та перетворення даних.

## 2. АРХІТЕКТУРА ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ УПРАВЛІННЯ ЕНЕРГОЕФЕКТИВНІСТЮ ЕКОНОМІКИ РЕГІОНУ

2.1 Вибір принципів та формування вимог до компонентів комп'ютерних штучних нейронних мереж реального часу

Архітектури ШНМ реального часу повинні в повній мірі використовувати можливості НВІС-технології, враховувати вартість площі кристала, а також кількість вхідних і вихідних виводів. Число зовнішніх виводів НВІС обмежене рівнем технології та розміром кристалу. В основу побудови ШНМ реального часу пропонується покласти принципи, які дозволять зменшити вартість, терміни і розширити галузі їх застосування. Аналіз показує [10], що забезпечити дані вимоги можна при використанні таких принципів:

1. модульності, який передбачає розробку компонентів АЗ у вигляді функціонально завершених пристроїв (модулів), що мають вихід на стандартний інтерфейс;
2. узгодженості інтенсивності надходження даних з обчислювальною здатністю АЗ;
3. конвеєризації та просторового паралелізму обробки даних;
4. однорідності та регулярності архітектури ШНМ;
5. локалізації та спрощення зв'язків між елементами АЗ;
6. відкритості програмного забезпечення, що передбачає можливості нарощування та його вдосконалення, максимального використанням стандартних драйверів та програмних засобів;
7. спеціалізації та адаптації апаратно-програмних засобів до структури алгоритмів обробки та інтенсивності надходження даних;
8. програмованості архітектури шляхом використання репрограмованих логічних інтегральних мікросхем.



## 2.2 Вибір елементної бази та варіантів побудови компонентів комп'ютерних штучних нейронних мереж реального часу

Ефективність комп'ютерної реалізації алгоритмів ШНМ безпосередньо пов'язана з вибором засобів реалізації: програмних, мікропрограмних або апаратних [7, 8].

При програмній реалізації алгоритмів ШНМ обчислювальні процеси переважно розгортаються в часі з великим об'ємом пересилання інформації між оперативною пам'яттю і операційними пристроями. Програмні засоби є доступними для програміста, перед яким виникає задача мінімізації об'єму програм і часу їх реалізації при заданій точності обчислень. Вказані засоби характеризуються низькою швидкістю і гнучкістю з точки зору можливості модифікації та заміни алгоритмів.

Мікропрограмна реалізація обчислень передбачає їхнє розгортання як в часі, так і в просторі. При мікропрограмуванні є доступ до системи мікропрограм процесора, що забезпечується застосуванням постійної пам'яті, програмованих логічних матриць, а також оперативних запам'ятовуючих пристроїв, які використовуються в якості пам'яті мікропрограм. Прикладом мікропрограмної реалізації є реалізація алгоритмів ШНМ на базі однорідних обчислювальних середовищ (ООС). Процесор на базі ООС – це двовимірна регулярна матриця процесорних елементів (ПЕ), кожен з яких фізично зв'язаний входом-виходом з чотирма сусідами – зверху, знизу, ліворуч та праворуч [8]. Кожний ПЕ може виконувати набір бітових операцій перетворення інформації з вхідних каналів у вихідні. Процесор на базі ООС є універсальною системою, тобто в ньому можливо реалізувати довільну обчислювальну функцію. Бітовий рівень ПЕ та повна система комутації дозволяють реалізувати розпаралелення обчислень на найнижчому бітовому рівні. Це є суттєвою перевагою мікропрограмної реалізації алгоритмів ШНМ. Реалізація в повній мірі потенційних можливостей мікропрограмних засобів може бути досягнута лише при глибокому вивченні як

задачі, яка розв'язується, так і внутрішньої мови процесора. Мікропрограмні засоби реалізації алгоритмів ШНМ в порівнянні з програмними є більш швидкодіючими.

Успіхи в області інтегральної технології дозволяють все більше перекладати реалізацію алгоритмів ШНМ на апаратні засоби, які розгортають обчислення не тільки у часі, а й у просторі. Такі обчислення характеризуються введенням додаткового обладнання і відсутністю проміжного пересилання інформації в процесі обчислення, а також спрощенням функції місцевого управління. В основі структурної організації апаратних засобів лежить принцип адекватного апаратного відображення потокових графів алгоритмів ШНМ на комбінаційну матрицю, процесорні елементи якої реалізують функціональні оператори та з'єднані між собою відповідно з потоковим графом алгоритмом [9]. Синтезовані за таким принципом структури є алгоритмічними. В алгоритмічних структурах алгоритм виконується над вхідними даними при їх одноразовому проходженні через всі ПЕ. За режимами роботи такі структури діляться на синхронні та асинхронні. В асинхронних (однотактних) структурах обробка даних здійснюється без проміжних запам'ятовувань. Продуктивність таких структур визначається часом виконання найскладніших функціональних операторів  $\Phi_{ij}$  алгоритму, які лежать на шляху проходження даних. Кожна однотактна структура є послідовною з точки зору реалізації функціональних операторів  $\Phi_{ij}$ . Це є причиною обмеженої швидкодії та неефективного використання обладнання при обробці інтенсивних потоків даних у реальному часі. Тому для обробки потоків даних доцільно використовувати синхронні структури з конвеєрною реалізацією графів алгоритмів, яка передбачає суміщення у часі виконання функціональних операторів алгоритму над різними даними. Апаратні засоби дозволять з максимальною швидкістю реалізовувати алгоритми ШНМ, скоротити витрати на створення та експлуатацію програм і є доступними тільки розробникам архітектур ШНМ.

Потрібно відзначити, що всі види реалізації алгоритмів в безпосередньому вигляді зустрічаються досить рідко. На практиці в більшості випадків для

реалізації алгоритмів функціонування ШНМ використовуються комбіновані підходи з перевагою одного з перерахованих засобів. Переважання того чи іншого засобу визначається вимогами, які ставляться до ШНМ за швидкістю, габаритами, потужністю споживання та ціною.

Розглянемо три основні варіанти побудови ШНМ реального часу:

1. на основі універсальних і функціонально-орієнтованих мікропроцесорів шляхом розробки спеціалізованого програмного забезпечення;

2. на основі універсального обчислювального ядра доповненого базовими апаратно-програмними компонентами, які реалізують часомісткі алгоритми функціонування ШНМ;

3. у вигляді спеціалізованої алгоритмічної системи, архітектура та організація обчислювального процесу в якій відображає структуру алгоритму розв'язання задачі.

Перший варіант є доступним для широкого кола користувачів. Істотною його перевагою є можливість використання раніше розроблених програм. Його недоліками є невисока швидкість, функціональна і структурна надлишковість комп'ютерних засобів.

Другий варіант є перспективним, оскільки він передбачає поєднання універсальних і спеціальних засобів. Таке поєднання забезпечує високу ефективність використання обладнання при створенні систем для опрацювання у реальному часі потоків даних за алгоритмами, які є нерегулярними з великою кількістю логічних операцій. При цьому розробка апаратних засобів з заданими технічними параметрами зводиться до доповнення обчислювального ядра додатковими апаратно-програмними компонентами.

Третій варіант орієнтований на обробку у реальному часі інтенсивних потоків даних за складними алгоритмами. При цьому висока ефективність використання обладнання досягається узгодженням обчислювальної здатності апаратних засобів з інтенсивністю надходження потоків даних. Використання для побудови апаратних засобів обчислювальних полів на основі ПЛІС з динамічним

репрограмуванням відкриває нові можливості, які пов'язані з оперативним переналаштуванням ПЛІС на реалізацію конкретного алгоритму.

Другий і третій варіанти є найперспективнішими для синтезу ШНМ реального часу з високою ефективністю використання обладнання.

### 2.3 Система підтримки прийняття рішень для системи енергоуправління

Розробку ІАСУЕ регіону пропонується здійснювати на основі компонентно-орієнтованої технології та інтегрованого підходу, який охоплює методи та засоби аналітичної обробки даних, моделювання, прогнозування та прийняття рішень. Основними задачами ІАСУЕ регіону є збір, аналітичне та інтелектуальне опрацювання енергетичних даних, візуалізація результатів опрацювання та підтримка прийняття управлінських рішень. Розробка ІАСУЕ регіону базується на інтегрованих засобах збору даних, на сховищах і базах даних, телекомунікаційних і Web технологіях. Така система повинна забезпечувати збір, архівацію та попередню оцінку даних; інтеграцію різноманітних даних за допомогою баз і сховищ даних, доступ до яких здійснюється засобами Web-технологій; оперативну та інтелектуальну обробку даних, моделювання та прогнозування енергоефективності економіки регіону; візуалізацію енергетичних даних і результатів обробки у вигляді графіків і діаграм. Розробку ІАСУЕ регіону пропонується здійснювати за такими принципами:

1. модульності, який передбачає розробку компонентів у вигляді функціонально завершених модулів, що мають вихід на стандартний інтерфейс;
2. відкритості та сумісності;
3. використання комплексу базових проектних рішень.

Архітектура ІАСУЕ регіону наведена на рис.1, де TCP/IP – стек протоколу обміну, ETL – (Extract, Transform, and Load) виймання, перетворення та

завантаження, OLAP - (online analytical processing) інтерактивна аналітична обробка, ІАД – інтелектуальний аналіз даних.

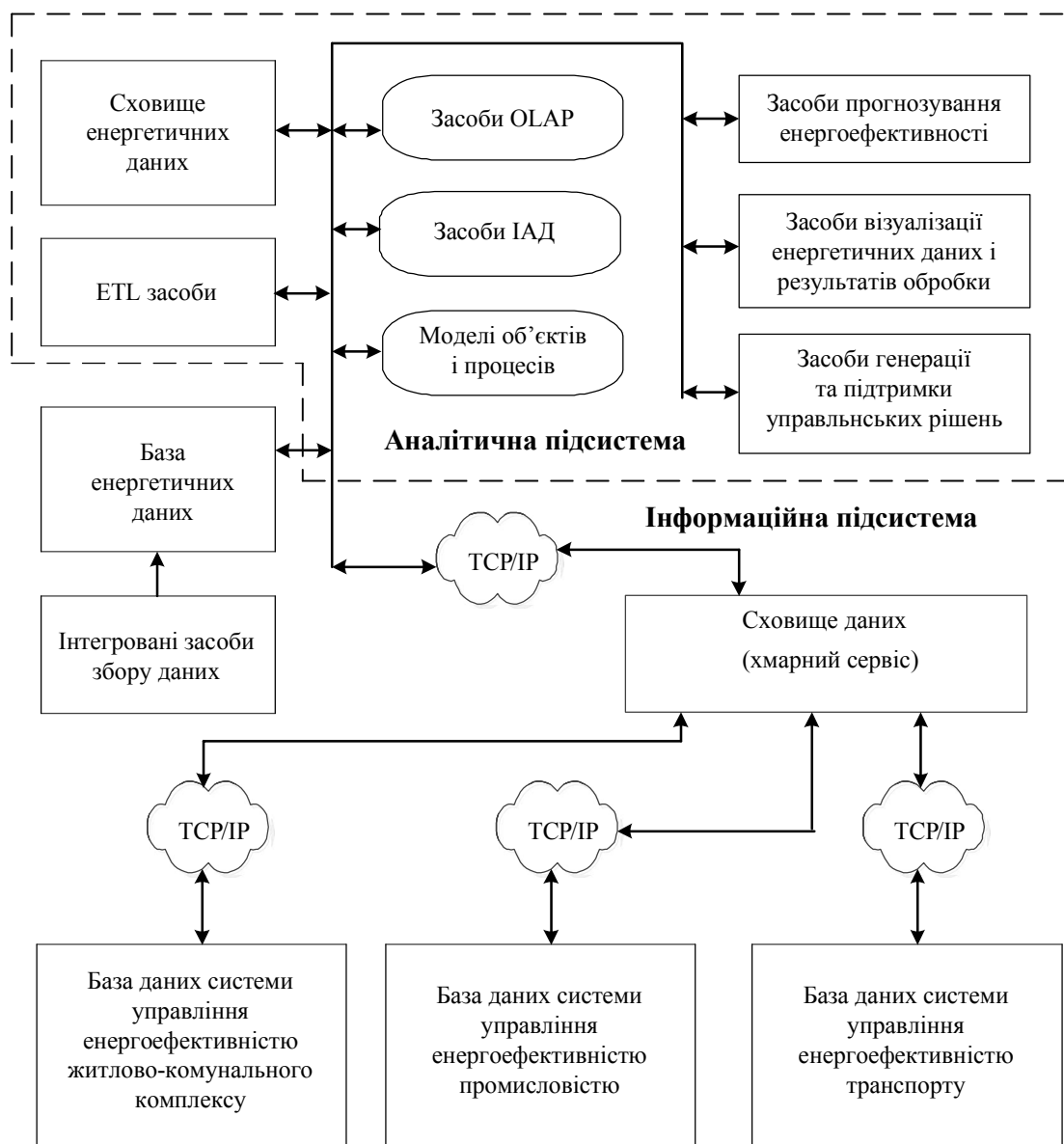


Рисунок 2.1 Архітектура ІАСУЕ регіону

Основними компонентами ІАСУЕ регіону є інформаційна та аналітична підсистеми. Інформаційна підсистема забезпечує збір енергетичних даних, які споживаються транспортом, промисловістю і житлово-комунальним господарством регіону. Технічною основою інформаційної підсистеми ІАСУЕ регіону є безпроводні сенсорні мережі, які забезпечують збір енергетичних даних. У загальному випадку бездротова сенсорна мережа – це розподілена система, яка

складається з малогабаритних інтелектуальних сенсорних пристроїв, які зв'язані з сховищем даних у хмарному сервері. Основними компонентами сенсорних мереж є просторово розподілені сенсори, засоби читання показників лічильників, обчислювані та передаючі блоки. Усі інтелектуальні сенсори взаємодіють через глобальну або локальну інфраструктуру інформаційного обміну з доступом до сховища даних у хмарному сервісі. Крім того, збір інформації про споживання енергоносіїв здійснюється користувачами шляхом внесення даних із статистичних збірників і сайтів (державних служб статистики України та Євростату, обліку енергоресурсів) у бази даних.

Інформація про споживання енергоносіїв транспортом, промисловістю та житлово-комунальним господарством зберігається у сховищі даних у хмарному сервісі, який має відкриті інтерфейси для доступу та публікації інтегрованих даних. Доступ до цих даних може здійснюватися будь яким користувачем мережі інтернет за допомогою публічних, документованих інтерфейсів доступу, які надає хмарний сервіс. За накопичення даних у хмарному сервісі безпосередньо відповідають підприємства та установи. Енергетичні дані з підприємств записуються у сховище даних у хмарному сервісі за допомогою інтерфейсів, які базуються на HTTP (Hyper Text Transfer Protocol) протоколі передачі гіпертекстових документів. Для роботи з сховищем даних у хмарному сервісі доцільно використовувати інтерфейс з HTTP REST протоколом, який є простим і уніфікованим інтерфейсом для роботи у мережі. Даний інтерфейс забезпечує користувачам можливість як напряму публікувати чи читати дані з сервісу, так і робити запити на керуючі дії. Прикладом таких дій може бути запит на встановлення сервісом потокового читання даних на основі TCP сокетів. Сервіс може надавати файлові інтерфейси для публікації файлів з даними.

Аналітична підсистема складається із сховища енергетичних даних, моделей об'єктів і процесів, засобів ETL, OLAP, ІАД, прогнозування, візуалізації, генерації та підтримки прийняття управлінських рішень. Сховище енергетичних даних будується на основі Big Data технологій, які забезпечують можливість ефективно зберігати, опрацьовувати та надавати доступ до великих об'ємів даних у

розподіленому режимі. Крім того, ці технології забезпечують ефективне опрацювання неструктурованих даних. Для реалізації і надання доступу до OLAP технології у сховищі використовуються OLAP сервери. Користувачі можуть отримати доступ до цих засобів як напряму через інтерфейси, так через програмні додатки (десктопні, веб чи мобільні). Пошук функціональних і логічних закономірностей, знаходження прихованих правил і закономірностей, побудова моделей і правил, які характеризують стан або прогнозують розвиток енергетичних процесів здійснюється засобами ІАД.

Інструменти підтримки прийняття рішень оцінюють енергоефективність економіки регіону, вибирають шляхи зменшення споживання палива та енергії, а також забезпечують виробничу та адміністративну підтримку. Аналіз та оцінка адміністративних рішень призводять до використання різних візуалізацій оперативної інформації для подальших адміністративних рішень. Якщо управлінські рішення прийматимуться з метою підвищення енергоефективності місцевої економіки, найкраще рішення буде знайдено серед багатьох прийнятних. Рішення цієї проблеми базується на даних про споживання енергії на регіональному транспорті, промисловості, житлово-комунальному господарстві, результатах прогнозу енергоефективності та регіональних стратегій економічного розвитку.

## 2.4 Аналіз елементної бази для апаратних реалізацій нейромереж

### 2.4.1 Каскадна архітектура.

Прикладом каскадної архітектури є NNP фірми Accurate Automation Corp. Він являє собою кілька 16-розрядних процесорів, що оснащені пам'яттю для зберігання вагових коефіцієнтів. Процесори зв'язані локальною внутрішньою

шиною і мають всього 9 команд. В комплект поставки входить бібліотека підпрограм з реалізованими нейромережними алгоритмами. Чіп поставляється на платах під шини ISA і VME. Продуктивність складає від 140 MIPS до 1,4 GIPS, в залежності від кількості нейрочіпів на платі. Їх кількість може бути від одного до десяти.

#### 2.4.2 RBF-архітектура

Мережі, що використовують радіальні базисні функції (RBF-мережі), є частковим випадком двошарової мережі прямого поширення. Кожний елемент прихованого шару використовує як активаційну радіальну базисну функцію типу гаусової. Радіальна базисна функція (функція ядра) центрується в точці, яка визначається ваговим вектором, зв'язаним з нейроном. Як позиція, так і ширина функції ядра повинні бути навчені за вибірковими зразками. Зазвичай ядер набагато менше, ніж навчальних прикладів. Кожний вихідний елемент обчислює лінійну комбінацію цих радіальних базисних функцій. З точки зору задачі апроксимації приховані елементи формують сукупність функцій, які утворюють базисну систему для представлення вхідних прикладів у побудованому на ній просторі. Існують різні алгоритми навчання RBF-мереж.

Прикладом RBF-архітектури є чіп Ni1000 (розробка фірми Intel разом з Nestor), в якому міститься близько 3 млн. транзисторів, що імітують 1024 нейрони, швидкодія 33 МГц, 17 GIPS. Випускається на платах у форматі ISA та VME.

#### 2.4.3 Систолічні нейропроцесори

Принцип дії систолічних процесорів полягає у використанні процесорної матриці з набору простих за структурою обчислювальних елементів, через яку йде неперервний потік даних, що змінюється кожним елементом. Мережа працює циклічно і в кожному наступному циклі кожний обчислювальний елемент зчитує і обробляє нову порцію даних, незалежно від роботи інших обчислювальних



елементів. Завдяки цьому досягається висока міра розпаралелення процесів обробки інформації і в результаті – висока швидкість роботи всієї мережі загалом.

Внаслідок значного спрощення внутрішньої структури обчислювальних елементів, систолічна матриця повинна бути оточена великою кількістю елементів, що реалізують додаткову функціональність. Наприклад, функція активації повинна бути реалізована у вигляді окремого зовнішнього блоку.

На рисунку 2.2 показана структурна схема систолічного процесора SAND (Simple Applicable Neural Device). Чіп розроблений Дослідним центром в Карлсруї та Інститутом мікроелектроніки Штутгарда.

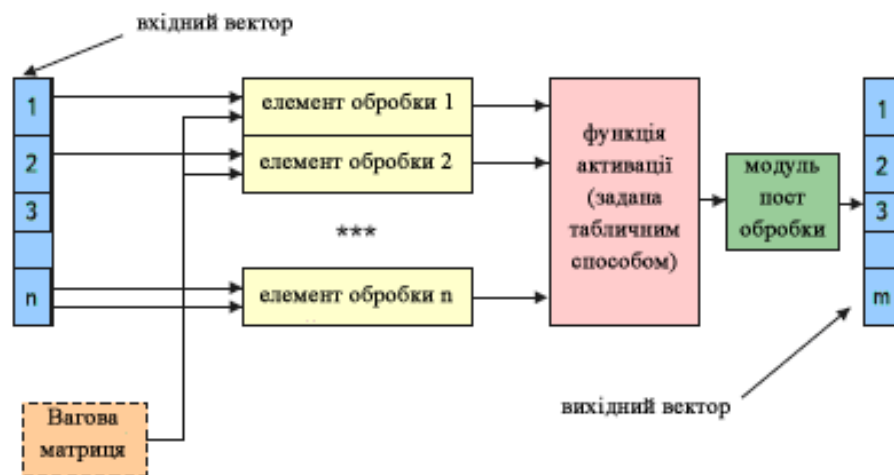


Рисунок 2.2. Структурна схема систолічного процесора SAND

Обладнання включає набір моделей наступної обробки та модель пам'яті. Комп'ютерна програма працює над розширенням та додаванням вхідних символів. Зовнішні логічні блоки використовуються для всіх інших функцій. У деяких випадках одиниці будуть враховуватися для першого пріоритету або першого зовнішнього значення.

#### 2.4.4 Сигнальні процесори

Відносно низька вартість розробки, численна елементна база з великою кількістю засобів відлагодження і допоміжного програмного забезпечення, економічне енергоспоживання і невелике тепловиділення, а також висока продуктивність зробили сигнальні процесори фірм Texas Instruments і Analog Devices найбільш популярними об'єктами для побудови апаратних реалізацій нейронних мереж. Моделювання нейронних мереж на сигнальних процесорах реалізується тільки програмним шляхом, що дозволяє при потребі легко модифікувати використовуваний алгоритм і структуру мережі.

Наприклад, використання розробниками DSP серії ADSP2116x компанії Analog Devices виправдано низкою вдалих конструктивних рішень, закладених в архітектуру цих чіпів. ADSP2116x використовують архітектуру SHARC (Super Harvard ARChitecture), яка передбачає розділення пам'яті і шин команд та даних, при цьому по шині команд можуть передаватися як команди, так і дані. Процесор має два генератори адреси даних, що дозволяє одночасно генерувати дві адреси для вибірки двох операндів.

Об'єм внутрішньої пам'яті розміром 1 Мбіт дозволяє реалізувати достатньо великі нейронні мережі. Система команд ADSP2116x містить арифметичну команду множення з накопиченням  $C=AxB+C$ , що досить зручно для побудови циклів обчислення вхідних значень нейронів. Мікропроцесор використовує циклічні буфери, при звертанні до яких відбувається автоматична модифікація адрес, що дозволяє спростити цикли і збільшити швидкість роботи мережі. Крім того, ADSP2116x використовує два обчислювальні блоки (обробних елементи, PE<sub>x</sub> та PE<sub>y</sub>) замість одного.

В стандартному режимі використовується тільки PE<sub>x</sub>. В режимі SIMD (Single Instruction, Multiple Data) процесор направляє одну і ту ж інструкцію в обидва процесорних елементи, завантажує два окремих набори даних з пам'яті в кожний PE, здійснює паралельні обчислення в обох PE і синхронізує результати в пам'яті. Таким чином, використання двох паралельно працюючих процесорних елементів дозволяє вдвічі збільшити швидкість обробки даних, що досить

актуально при моделюванні нейронних мереж, що складаються з великої кількості однотипних елементів.

#### 2.4.5 Нейросигнальні процесори

Основою нейросигнальних процесорів є те ж саме ядро, що й для інших розпоширених процесорів DSP. Відмінність полягає в додаткових архітектурних рішеннях, що оптимізують дані процесори для розробок на їхній основі нейронних мереж. В якості прикладу приведемо нейропроцесор NEUROMATRIX NM6403 фірми НТЦ "Модуль".

Цей процесор складається з основного 32-розрядного RISC-ядра і додаткового 64-розрядного векторного співпроцесора. NM6430 має стандартний інтерфейс вводу-виводу, що сумісний з портами TMS320C4, може працювати з числами змінної розрядності від 1 до 64 біт, його адресний простір досягає 16 Гбайт. Крім того, він має підтримку векторно-матричних і матрично-матричних операцій, а також підтримку двох типів функцій насичення на кристалі. Його продуктивність по скалярних операціях складає 120 MIPS (32-розрядні операнди) і від 40 до 11500 ММАС (мільйонів множень з накопиченням за секунду). Потрібно зауважити, що хоча процесор NM6403 завдяки оригінальній архітектурі значно краще справляється з матричними операціями, ніж інші DSP, але весь тягар програмної реалізації вибраної нейронної мережі як і раніше лежить на розробнику.

#### 2.4.6 ПЛІС як елементна база нейронних мереж

В останні роки стрімко зросла зацікавленість до використання ПЛІС в якості елементної бази нейрообчислювачів. Причина цього полягає у появі на ринку високоякісних ПЛІС, продовження тенденції до зниження їхньої ціни, і, відповідно, і кінцевої вартості розробок. Напруга живлення ПЛІС з кожним новим поколінням цих мікросхем постійно зменшується, що дозволяє зменшувати енергоспоживання при збільшенні швидкодії. Сучасні ПЛІС виготовляються такими відомими виробниками, як Altera, Atmel, Xilinx та іншими.

Завдяки технологічним характеристикам FPGA (FPGA) вони є зручним об'єктом для використання в якості базової основи для застосування обладнання до нейронних мереж. При розробці конструкцій на ПЛІС використовується принцип однобітної синхронізації, який реалізований в ПЛІС, яка розподіляє

тактові сигнали по високошвидкісній деревці мережі, заохочуючи майже одночасний прихід тактового імпульсу на всі внутрішні тактові входи. Тактовий сигнал може формуватися в самій ПЛІС.

Ця властивість полегшує завдання побудови багатосторонніх конфігурацій нейронних мереж. Сучасні ПЛІС практично неможливо програмувати "вручну". Для розробки проектів використовуються спеціальні інтегровані середовища програмування та паралельні мови програмування (VHDL). Це дозволяє усунути залежність розробок нейромереж FPGA від певних особливостей дизайну FPGA, покращити портативність та побудувати бібліотеки стандартних нейромережевих модулів, вигідно диференціюючи конструкції нейронних мереж на FPGA від подібних розробок на RCS. .

Нарешті, ще однією важливою особливістю ПЛІС, які дозволяють розвивати нейронні мережі на основі цієї основної передумови, є те, що всі ПЛІС містять багато подібних елементів, що підвищує надійність мікросхеми та дизайн нейронної мережі. .

Однак проектування нейронних мереж на ПЛІС характеризується відносно високим обладнанням затрати. Якщо врахувати, що для реалізації десяти паралельно працюючих нейронів потрібно задіяти декілька тисяч вентилів, і при цьому кожний нейрон потребує як мінімум одного помножувача, можна зробити висновок, що існуючі сьогодні на ринку ПЛІС з інтеграцією в декілька мільйонів вентилів найбільш придатні для реалізації малих і середніх за розміром нейронних мереж порядку кілька тисяч нейронів, і не придатні для створення великих і надвеликих нейронних мереж, що містять сотні тисяч чи мільйони нейронів.

Прикладом реалізації нейронної мережі на ПЛІС може бути нейромодуль цифрової обробки сигналів XDSP-680-CP фірми Scan Engineering Telecom.

## 2.5 Вибір засобів для візуалізації енергетичних даних

Мультимедійне відображення даних – представлене в форматі графіки та діаграм, інфографіки, схем, дашбордів, карт і картограм. Графіково діаграмний формат – найбільш звичний засіб візуалізації даних, які використовується для презентування та аналізу. Значна кількість загальновідомих типів діаграм, а їх

кількість варіативно росте – розробляються і додаються нові інструменти для зображення складних і незвичайних даних. Інфографіка показує варіації параметрів, стосунками між елементами. Презентація та опрацювання даних – один із поширених методів застосування візуалізації даних у графово-діаграмному форматі. При аналізі даних за методикою мультимедійного отримання візуального представлення даних з метою знаходження прихованих взаємозв'язків і впливів, а також, перш прийдешнього опрацювання з метою застосування в подальшому більш складних інструментів аналізу. Такий метод є Exploratory data analysis (EDA), тобто EDA є одним із інструментів Data Mining. Основою відмінності є EDA від візуалізації та представлення даних є те, що це буде за допомогою різноманітних програмних засобів.. Бізнес аналіз і дошки даних сьогодні активно використовуються коли необхідно стежити за певними критеріями на постійній основі.

Ця дошка забезпечує унаочнення подання результатів, які посортовані за темами для полегшення розуміння інформації. Розвиток даної функції дашбордів пов'язаний із застосуванням аналітичних методів в бізнесі. Картографіка одна з найдавніших способів візуалізації для відображення оточуючої реальності. Картограма це нанесена інформація у вигляді маркування кольорами або іншим способом. Картограми можуть бути використані для відображення рівня споживання енергоносіїв в визначеному місці регіону. Для створення картографічних документів можливе використання геоінформаційних систем (ГІС) реалізованих у вигляді програмних додатків для роботи з наборами просторової інформації на комп'ютері. ГІС як інформаційна система, призначена для збирання, аналізу та візуалізації просторових даних. Просторова інформація, класифікована певним чином, структурується і зберігається в спеціалізованих базах даних. ГІС обробляє не тільки дискретні дані, але і інформацію про безперервні явища, подану растрами, матрицями, іншими моделями даних. Сучасні ГІС мають здебільшого модульний склад, коли певна частина операцій включається до базового складу системи, а інші додаються при необхідності. Просторова складова становить сьогодні значну частину всіх даних, з якими

взаємодіють організації та установи. Сучасні ГІС вийшли за рамки поняття системи, що обробляє власне просторові дані, адже вони дозволяють працювати не тільки з картографічним матеріалом та атрибутами об'єктів на них, але і з різноманітними типами документів (текстових, графічних, мультимедійних), пов'язаних з певними об'єктами, забезпечувати виконання складних запитів до баз даних та перетворювати їх результати у картограми чи діаграми, прив'язані до певних територій тощо. Серед них не тільки обробка матеріалів польових вимірювань та спостережень, але і оформлення їх у вигляді карт та схем, зберігання картографічних даних різних типів та відображення окремих картографічних даних та різних комбінацій даних. При цьому забезпечується пошук даних за їх атрибутами, розташуванням відносно заданого об'єкту чи групи об'єктів, аналіз знаходження об'єктів, топологічних відношень, наявності та щільності розподілу об'єктів. аналіз атрибутів об'єктів карт, класифікація даних. Сучасні мережеві ГІС забезпечують аналіз та відображення змін даних у часі, побудову графових структур, мережевий аналіз, вирішення транспортних задач, вирішення задач проектування об'єктів та територій.

У системах управління енергоефективністю економіки регіону ГІС використовуються для керування шарами даних з датчиків первинної інформації, інформації про промислові об'єкти та установи, показники енергоспоживання та відповідної енергоефективності тощо. На відміну, спеціалізовані ГІС вирішують завдання лише певного напрямку та мають спеціалізований набір інструментів, що краще задовольняє потреби користувачів, яким потрібно вирішувати певне обмежене коло завдань. Такі ГІС створюються на платформі універсальних ГІС або як самостійні системи та вимагають наявності спеціалізованого програмного забезпечення, однак, як правило, недоступні у вигляді мережевих веб-додатків.

Геодані поділяються на два основних типи: растрові та векторні. Географічні дані сітки складаються з географічного позначення растрового зображення, такого як супутникові зображення. На додаток до супутникових знімків, растр використовується для цифрових моделей місцевості, де кожен піксель області зображення містить інформацію про висоту в цій точці області.

Векторні геодані описуються серією або послідовністю координат, геометрії та характеристичних значень: точки - для визначення точкової ознаки, яка є достатньою парою координат, приклад точкових ознак на карті - POI (цікаві точки); лінії: геометрія лінії визначається низкою пар координат; багатокутники: визначаються координати вершини. Окрім типу геометрії та розташування, атрибутивна інформація є менш важливою. У векторних даних кожен об'єкт може містити додаткову інформацію про атрибути. Ці атрибути дозволяють вибирати об'єкти та застосовувати до них різні правила обробки. Більшість інструментів ГІС мають можливість переглядати дані у табличному форматі, але для векторних геоданих розроблено широкий спектр форматів. Наприклад, Shapefiles - це формат, що використовується для ГІС-пакетів Esri, але він виявився простим у використанні і зараз є стандартом для інших програм географічної інформації. KML (Keyhole Markup Language) - це мова розмітки на основі XML для географічних даних. Додаток Google Earth - найдоступніший інструмент для роботи з географічними даними в Інтернеті, а файли KML популярні в картографічних сервісах. GPX - це текстовий формат на основі XML, який активно використовується для запису GPS-слідів. CSV - завдяки простоті загального текстового формату для зберігання геоданих координати задаються в стовпцях таблиці, CSV в основному використовується для об'єктних точок. GeoJSON - текстовий формат, завдяки зручності цього формату в JavaScript, останнім часом активно використовується для інтерактивного картографування. GeoJSON базується на JavaScript Object Notation (JSON). GeoJSON призначений для представлення різних структур географічних даних. Об'єкт GeoJSON може представляти геометрію, об'єкт або набір об'єктів. У GeoJSON об'єкт - це набір пар ключ-значення (властивості). Ім'я кожної властивості має тип стрічки, а значенням властивості може бути тип стрічки, номер, об'єкт, масив або одна з букв. Об'єкт GeoJSON може мати будь-яку кількість властивостей (пари ключ-значення).

Існуючі програми картографування допомагають швидко і легко візуалізувати існуючі дані, але насправді бувають випадки, коли дані повинні

оброблятися для візуалізації на основі конкретних потреб бізнесу. У цьому випадку слід використовувати існуючі мережево-орієнтовані програмні засоби з великою кількістю даних та веб-сервісів. Мовні засоби програмування Python здатні обробляти великий набір даних, що особливо корисно для комп'ютерного аналізу. Мова Python має легкий для розуміння синтаксис. Інструменти PHP орієнтовані на Інтернет, стандартний інсталяційний пакет включає досить гнучку графічну бібліотеку для PHP під назвою CD, що дозволяє створювати зображення та обробляти наявні зображення з нуля. Графічна бібліотека Sparkline дозволяє додавати візуальні компоненти в документ. PHP підключений до бази даних, наприклад MySQL, що дозволяє працювати з великими наборами даних. З іншого боку, обробка мови програмування пропонує просте середовище програмування, можливість створювати динамічні та інтерактивні діаграми з кількома рядками коду за допомогою різних бібліотек. У більшості випадків інтерактивна та динамічна графічна інформація в мережі розробляється за допомогою Flash та ActionScript. Графіку можна створити у Flash, а ActionScript допоможе вам контролювати взаємодію користувачів з інтерактивними об'єктами. Програми Flash завантажуються відносно швидко, і на більшості комп'ютерів вже встановлено програвач Flash Player за замовчуванням.

Аналіз показує, що існують загальнодоступні та доступні способи взаємодії з Інтернет-ресурсами: веб-браузери постійно покращують свою продуктивність, а інструменти HTML, JavaScript та CSS стають досить ефективними при відображенні даних. Нещодавно була покращена візуалізація, яку легко реалізувати у браузері цими способами. Якщо ви раніше створювали графіку з інтерактивними елементами, переважно на Flash та ActionScript, існують надійні пакети та бібліотеки, які допомагають швидко створювати інтерактивні та статичні візуальні ефекти, які можна використовувати для роботи з користувацькими даними. Використовувати JavaScript набагато простіше за допомогою бібліотек jQuery або MooTools. Плагіни бібліотеки допомагають створювати деякі основні типи діаграм та графіків. Перевага полягає в тому, що немає необхідності встановлювати графічну бібліотеку PHP на сервері.



## 2.6 Висновки до розділу

Запропоновано нейромережеві засоби будувати за такими принципами: змінного складу обладнання; модульності, який передбачає реалізацію компонентів у вигляді функціонально-завершених пристроїв, що мають вихід на стандартний інтерфейс; узгодженості інтенсивності надходження даних з обчислювальною здатністю апаратних засобів; конвеєризації та просторового паралелізму обробки даних; локалізації та спрощення зв'язків між елементами компонентів; спеціалізації та адаптації апаратно-програмних засобів до структури алгоритмів та інтенсивності надходження даних.

## 3 СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ КОНТРОЛЮ ЕНЕРГОМЕРЕЖІ

### 3.1 Реалізація апарату нейронних мереж

#### 3.1.1 Етапи синтезу апаратних засобів.

Синтез АЗ складається із наступних етапів: вибору та розробки методів і алгоритмів узгоджено-паралельної обробки, визначення основних параметрів та переходу від алгоритму до структури [14].

При виборі та розробці методів і алгоритмів узгоджено-паралельної обробки враховуються вимоги  $R$  і характеристики  $H$ , але визначальним є забезпечення обмежень  $B$ . Для оцінки розроблених алгоритмів використовуються характеристики: інформаційні, операційні та точності. До інформаційних характеристик відносяться: кількість констант, вхідних, вихідних і проміжних даних, кількість каналів і їх розрядність, кількість і види операцій. Операційні характеристики дозволяють оцінити час реалізації та обчислювальну здатність. До характеристик точності алгоритму відносяться: розрядність операційних пристроїв, способи округлення. У інформаційних технологіях узгоджено-паралельної обробки сигналів у реальному часі необхідно забезпечити взаємозв'язок і узгодження параметрів апаратних засобів. Для апаратних засобів реального часу одним із найважливіших параметрів є узгодженість інтенсивності надходження даних із обчислювальною інтенсивністю на всіх етапах обробки.

До основних параметрів оцінки апаратних засобів реального часу, крім витрат обладнання, швидкодії, ефективності використання обладнання пропонується використовувати обчислювальну здатність. Для асинхронних (однотактних) апаратних засобів, обробка даних в яких здійснюється без проміжних запам'ятовувань, обчислювальна здатність визначається так:

$$D_o = \frac{m_o n_o}{\sum_{j=1}^h \max_k t_{BO_k}} \quad (3.1.1)$$

де  $m_o$  – кількість каналів надходження даних;  $n_o$  – розрядність каналів;  $h$  – кількість послідовно з'єднаних операційних блоків,  $t_{BO_k}$  – час реалізації операційними блоками найскладніших функціональних операторів  $\Phi_{jk}$ .

Кожна однотактна структура є послідовною з точки зору реалізації функціональних операторів  $\Phi_{jk}$ . Це є причиною обмеженої швидкодії та неефективного використання обладнання при обробці інтенсивних потоків даних у реальному часі. Тому для обробки потоків даних доцільно використовувати синхронні структури з конвеєрною реалізацією графів алгоритмів, в яких здійснюється суміщення у часі виконання функціональних операторів алгоритму над різними даними. Конвеєризація структур апаратних засобів передбачає розділення їх на сходинки шляхом введення буферної пам'яті. При цьому, кожна сходинка конвеєра складається з двох компонентів: буферної пам'яті та операційних пристроїв, які реалізують функціональні оператори ярусу. Для забезпечення високої швидкодії та ефективності використання обладнання функціональні оператори, які реалізуються у сходинках конвеєра, мають бути простими та мати приблизно однаковий час реалізації. Однотактні алгоритмічні пристрої можна розглядати як одноступінчатий конвеєр. У зв'язку з цим актуальним є розгляд питань пов'язаних із синтезом конвеєрних структур апаратних засобів реального часу з високою ефективністю використання обладнання.

У конвеєрних АЗ обчислювальна здатність визначається так:

$$D_K = \frac{m_K n_K}{t_{БП} + t_{ОБ}}, \quad (3.1.2)$$

де  $t_{БП}$  – час звертання до буферної пам'яті,  $t_{ОБ}$  – час обчислення операційним блоком найскладнішого функціонального оператора  $\Phi_{jk}$ ,  $m_K$  –

кількість каналів надходження даних у сходинках конвеєра,  $n_k$  – розрядність каналів надходження даних у сходинках конвеєра.

### 3.1.2 Шляхи підвищення ефективності використання обладнання

В АЗ для ШНМ реального часу висока ефективність використання обладнання досягається мінімізацією витрат обладнання при забезпеченні реального часу. Перехід від алгоритму розв’язання задачі в реальному часі до структури АЗ формально зводиться до мінімізації витрат обладнання

$$W_{AZ} = W_{ПУ} + W_{ІІІ} + W_{ІІ} + \sum_{i=1}^k W_{ІІЕ} \cdot m_i \quad (3.1.3)$$

де  $W_{AZ}$ ,  $W_{ПУ}$ ,  $W_{ІІІ}$ ,  $W_{ІІ}$ ,  $W_{ІІЕ}$  – витрати обладнання відповідно на АЗ для ШНМ, пристрої управління, інтерфейсні пристрої, пам’ять,  $k$  – кількість типів процесорних елементів,  $i$ -ий процесорний елемент,  $m_i$  – кількість процесорних елементів  $i$ -о типу, при забезпеченні такої умови:

$$\frac{Nn}{F_d m_d n_d} = \frac{t_k Nn}{m_{ee} n_{ee}} \quad (3.1.4)$$

Основними шляхами мінімізації АЗ для ШНМ реального часу є:

1. врахування величини зміни елементів даних;
2. вибір ефективних методів і алгоритмів реалізації функціональних операторів;
3. зменшення розрядності операційних пристроїв, пам’яті, кількості і розрядності каналів передачі даних;
4. узгодження інтенсивності надходження даних із обчислювальною здатністю апаратних засобів.

### 3.1.3 Розробка високоефективних багат шарових перцептронів з неітераційним навчанням

Для реалізації розроблених алгоритмів навчання та функціонування БШП будемо використовувати універсальні та спеціалізовані обчислювальні засоби, які об’єднуються в проблемно-орієнтовану обчислювальну систему. В якості

універсальних засобів будемо використовувати функціонально-орієнтовані процесори, а в якості спеціалізованих – алгоритмічну систему, архітектура та організація обчислювального процесу в якій відображає структуру алгоритму розв’язання задачі [11]. Побудову такої алгоритмічної системи доцільно здійснювати на базі обчислювального поля, елементною базою якого є ПЛІС з динамічним репрограмуванням. Використання такої елементної бази відкриває нові можливості, які пов’язані з оперативним переналаштуванням ПЛІС на реалізацію конкретного алгоритму. Структура проблемно-орієнтованої обчислювальної системи, яка реалізує алгоритми навчання та функціонування БШП з неітераційним навчанням наведена на рисунку 3.1, де ПП – паралельна пам’ять, БПП – буферна паралельна пам’ять, ФОП – функціонально-орієнтований процесор.

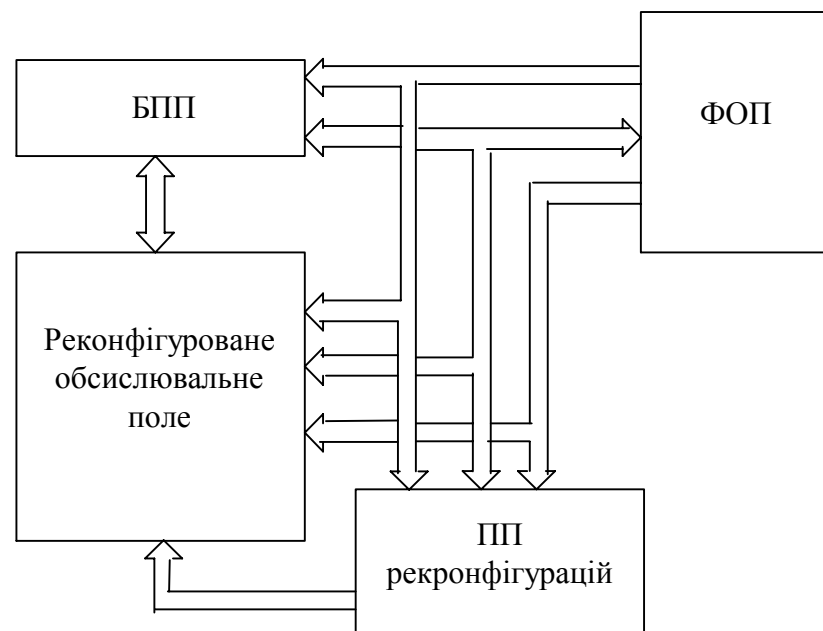


Рисунок 3.1 Структура проблемно-орієнтованої обчислювальної системи для реалізації БШП з неітераційним навчанням

Основними компонентами розробленої системи є: буферна паралельна пам’ять і паралельна пам’ять реконфігурацій, функціонально-орієнтований процесор ФОП і реконфігуроване обчислювальне поле. Буферна паралельна

пам'ять призначена для зберігання, упорядкування та комутації вхідних даних, а паралельна пам'ять реконфігурацій призначена для зберігання даних налаштування реконфігурованого обчислювального поля на реалізацію компонентів узгодженого графу БШП.

Архітектура ФОП повинна бути орієнтована на реалізацію алгоритмів навчання та функціонування БШП. В запропонованій проблемно-орієнтованій обчислювальній системі ФОП призначений для виконання попередніх обчислень, пов'язаних з формуванням макрочасткових добутоків, здійснення налаштування реконфігурованої обчислювальної системи на реалізацію конкретних алгоритмів та управління обчислювальним процесом.

Реконфігуроване обчислювальне поле програмно налаштовується на апаратне відображення компонентів узгодженого потокового графу алгоритму навчання та функціонування БШП, де кожному НЕ ставиться у відповідність операційні пристрої, які забезпечують обчислення оператора суми парних добутоків і виконання активаційної функції. Для апаратного відображення оператора суми парних добутоків використовуються розроблені структури, інтенсивність обчислення в яких узгоджена з інтенсивністю надходження даних. Використання реконфігурованого обчислювального поля в проблемно-орієнтованій обчислювальній системі забезпечує оперативне налаштування програмним шляхом даного поля на реалізацію компонентів узгодженого потокового графу алгоритму навчання та функціонування БШП.

#### 3.1.4 Оцінка основних параметрів базових компонентів

Основними елементами розроблених структур пристроїв для реалізації базових операцій ШНМ є функціональні вузли, затрати обладнання та швидкодія яких наведена у таблиці 3.1 [42] Оскільки розроблені структури орієнтовані на НВІС-реалізацію, то за одиницю вимірювання витрат обладнання в взято логічний вентиль, який являє собою елемент типу інвертор, І, АБО, а для оцінки часових параметрів - величину затримки логічного вентиля  $\tau$ .

Таблиця 3.1 Затрати обладнання та швидкодія функціональних вузлів

п/п	Найменування функціональних вузлів	Витрати обладнання (вентилів)	Кількість каскадів затримки ( $t$ вентилів)
.	тригер	$\beta 6$	$\lambda 3$
.	регістр	$\beta 7n$	$\lambda 3$
.	однорозрядний суматор	$\beta 18$	$\lambda 7$
.	однорозрядний віднімач	$\beta 18$	$\lambda 7$
.	однорозрядний суматор-віднімач	$\beta 20$	$\lambda 8$
.	$n$ -розрядний суматор	$\beta 20n$	$\lambda 7 \log_2 n$
.	$n$ -розрядний віднімач	$\beta 21n$	$\lambda 8 \log_2 n$
.	$n$ -розрядний суматор-віднімач	$\beta 23n$	$\lambda 8 \log_2 n$
.	$m$ -вхідний $n$ -розрядний суматор	$\beta (m-1) 20n$	$\lambda 7 \log_2 n \log_2 m$
0.	$m$ -вхідний $n$ -розрядний конвеєрний суматор	$\beta 27 (m-1)n$	$\lambda 10$
1.	перемножувач	$\beta 18 n_2$	$\lambda 14 n$
2.	квадратор	$\beta 9 n_2$	$\lambda 12 n$
3.	пристрій ділення	$\beta 20 n_2$	$\lambda 16 n$
4.	схема порівняння	$\beta 7 n$	$\lambda 3 \log_2 n$
5.	двійковий лічильник	$\beta 12 n$	$\lambda 5 \log_2 n$
6.	дешифратор $m \times l$	$\beta (2m+2\log_2 l)$	$\lambda m$
7.	$m$ -вхідний $n$ -розрядний комутатор	$\beta 3mn$	$\lambda m$
8.	$m$ -вхідний $n$ -розрядний ПЗП	$\beta 2_m n$	$\lambda (m+3)$
9.	$m$ -вхідний $n$ -розрядний ОЗП	$\beta 2_m 3n$	$\lambda (m+3)$

Для оцінки характеристик розроблених пристроїв для реалізації базових операцій ШНМ використаємо дані про витрати та величину затримки взяті з таблиці 3.1. На основі цих даних для пристрою для вертикально-паралельної

реалізації оператора суми парних добутоків (рис. 5.1) розробимо аналітичні вирази для розрахунку витрат обладнань та часу обчислень у залежності від розрядності та кількості чисел.

$$W_{оспд} = n \cdot m \cdot W_{T_2} + n \cdot W_{T_2} + n \cdot m \cdot W_i + n(W_{БСМ} + W_{P_2} \log_2 m + W_{См} \log_2 m + W_{P_2}(n + \log_2 m) + W_{Шф}) + W_{См}(\log_2 m + n) + W_{P_2}(2n + \log_2 m);$$

$$W_{оспд} = 6nm + 6n + nm + n(20(m - 1) + 7 \log_2 m + 18 \log_2 m + 7(n + \log_2 m) + 3(n + \log_2 m)) + 18(n + \log_2 m) + 7(2n + \log_2 m).$$

$$W_{оспд} = 10n^2 + 18n + 27nm + 35n \log_2 m + 25 \log_2 m$$

На основі аналітичних виразів, наведених вище, була розроблена програма в середовищі Embarcadero RAD Studio, яка будує таблицю витрат  $W$  обладнання у залежності від кількості операндів  $m$  та їхньої розрядності  $n$  і, також, малює діаграму на основі даних обчисленої таблиці. Текст програми подано у додатку А.

Приклад реалізації програми наведено на рисунку 3.2.

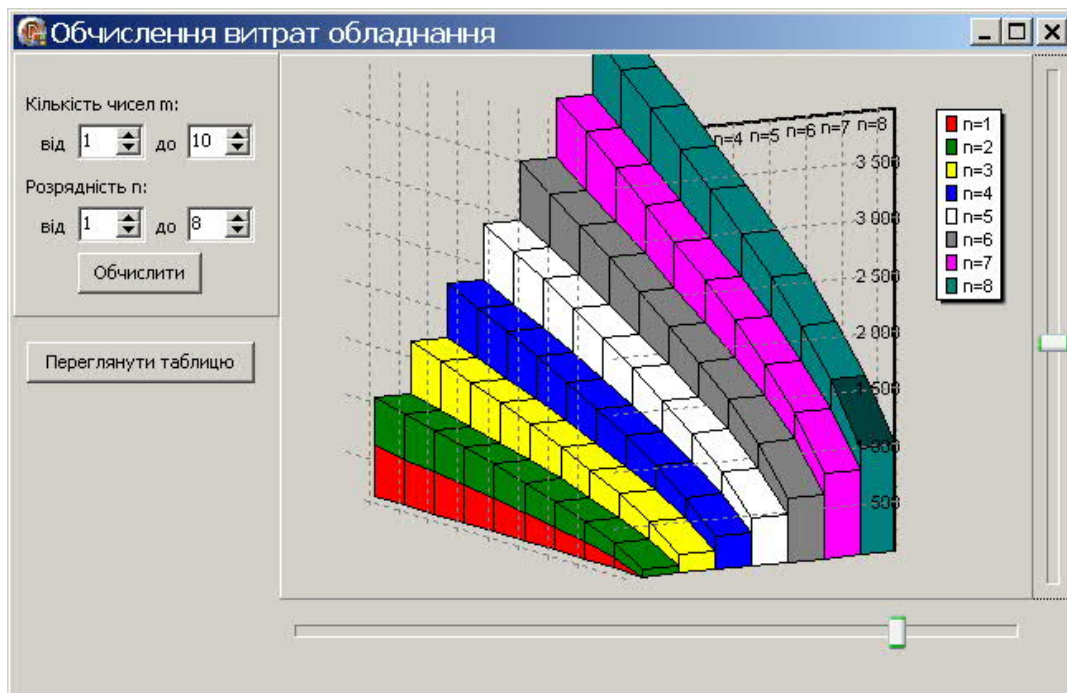


Рисунок 3.2 Приклад реалізації програми

З аналізу даних, отриманих під час обчислення за допомогою програми, можна зробити висновок, що величина витрат обладнання при вертикально-



паралельному обчисленні оператора суми парних добутків більшою мірою залежить від розрядності вхідних операндів, а не від їхньої кількості.

### 3.2 Розроблення засобів візуалізації просторових енергетичних даних

Дані датчиків, що використовуються в ГІС, зазвичай мають географічне (просторове) посилання. Дані мають префікс "гео" з моменту, коли вони з'єднуються з місцевістю і відображаються на карті. Об'єкт пов'язаний із землею за допомогою географічних координат (широти та довготи) та, за бажанням, тривимірного зображення - висоти. Дані датчиків та інша інформація є непросторовими, тому функція ГІС полягає у встановленні зв'язку з просторовими даними. ГІС може зберігати набір параметрів для кожного об'єкта. За допомогою програми ГІС ви можете змінити вигляд карти на основі непросторової інформації, пов'язаної з розташуванням певного датчика (об'єкта), відповідно до його просторового зв'язку.

Зазвичай на невеликій території знаходиться значна кількість різних об'єктів. Вони мають різні типи локалізації. Важливою особливістю ГІС є можливість класифікації об'єктів з різних причин, що важливо для завдань цієї системи, тому інформація в ГІС поділяється на певні теми. У межах теми об'єкти поділяються на шари (відповідно до видів енергії, що використовується, відповідно до класифікації аксесуарів тощо). Служба даних у хмарній службі має відкриті інтерфейси для доступу та публікації інтегрованих даних, керованих хмарною службою. Будь-який користувач Інтернету може отримати доступ до цих даних через задокументовані інтерфейси загального доступу, надані хмарною службою. Компанії та установи несуть безпосередню відповідальність за накопичення даних у хмарній службі. Бізнес-дані записуються в базу даних у хмарній службі за допомогою інтерфейсів на основі протоколу HTTP. Зберігання

даних можна реалізувати в хмарній службі, використовуючи найпопулярніші хмарні сервіси, такі як: AWS, Azure та Google Cloud.

Використані інструменти ГІС допомагають візуалізувати зібрані дані відповідно до конкретних вимог до діяльності. ГІС може працювати з базами даних для пошуку та відбору даних про енергоефективність. Просторова інформація, що зберігається у спеціалізованих базах даних, забезпечує обробку зібраної інформації відповідно до завдань. Компоненти розробки оперативної, аналітичної та інтелектуальної обробки даних використовують геоформовані параметри асоціації для забезпечення візуалізації космічних даних. ГІС-аналізатор, що реалізує алгоритми та завдання просторової обробки даних, забезпечує підтримку в управлінських рішеннях на рівні окремої організації чи установи та на районному, регіональному чи регіональному рівнях. Цей підхід не тільки реалізує обробку окремих вимірювальних матеріалів та спостереження за об'єктами, але також передбачає формування інтегрованих показників у вигляді карт і діаграм, а також зберігання картографічних даних. Забезпечують візуалізацію даних на основі їх характеристик, їх положення щодо певної групи об'єктів енергоефективності, щільності розподілу об'єктів, атрибутів об'єктів чи їх класифікації з урахуванням змін даних у часі, реалізує формування управлінських рішень щодо енергоефективності на основі отриманої таким чином об'єктивної інформації про стан енергоспоживання.

Для побудови компонентів візуалізації даних енергоспоживання в ГІС використовується інтерфейс API (Application Programming Interface), з допомогою якого програмні компоненти взаємодіють один з одним. При цьому зазвичай компоненти утворюють ієрархію – високорівневі компоненти використовують API низькорівневих компонентів. Стратегія API включає розробку компонентів на основі існуючих API, надання внутрішніх сервісів розробникам. Стандартні API для забезпечення функціональності відіграють важливу роль, оскільки вони гарантують, що всі компоненти будуть працювати однаково. У випадку API графічних інтерфейсів це означає, що програми будуть мати схожий користувацький інтерфейс, що полегшує процес освоєння нових програмних

продуктів. У випадку веб-додатків, застосування стандартизованих API забезпечує кросплатформну сумісність функціонування додатків при застосуванні веб-браузерів, реалізує передачу інформації і забезпечує взаємодію за допомогою стандартного HTML формату. Загальнодоступні API забезпечують передачі даних в одному з двох форматів: XML або JSON. JSON набагато більш лаконічний і простий у сприйнятті людиною, ніж XML.

Інтерфейс програмування додатків Google Maps API забезпечує створення необхідного користувачу інтерактивного картографічного матеріалу ГІС системи. Цей сервіс надає доступ до величезної картографічної бази даних – традиційних карт, супутникових знімків земної поверхні, панорам перегляду вулиць та використовує тривимірні зображення і моделі. Картографічна база постійно оновлюється, що забезпечує можливість максимально точно відобразити реальний стан об'єктів, у тому числі і енергоспоживання. Картографічний ресурс складається з бази даних дистанційного зондування і топографічних карт. Для доступу до даних пропонується два сервіси – Google Maps і Google Earth. Google Maps сервіс <http://maps.google.com/> працює в режимі веб-додатку, де доступ до даних максимально спрощений. Для перегляду достатньо програмного забезпечення веб-браузера. Зручність навігації обумовлена використанням технології AJAX, що дозволяє оновлювати вміст сторінки без її перезавантаження. Таким чином здійснюється зміна масштабу і переміщення по карті без будь-яких затримок. Google Earth сервіс <http://earth.google.com/> працює з даними за допомогою спеціальної програми-навігатора, що вимагає окремої установки. Можливості програми перевершують веб-сервіс як набором даних, так і наявністю додаткових інструментів (вимірювання, робота з GPS). Використання Google Earth забезпечило популяризацію формату KML (Keyhole Markup Language), що представляє собою розширення XML розроблене для зберігання і опису векторних просторових даних. Популярність формату забезпечує імпорт та експорт в найпоширеніші пакети ГІС та конвертування даних з інших форматів в KML. За допомогою сервісу Google Maps API (Google Maps Engine) створюються призначені для користувача карти з мітками, шарами, маршрутами. Компоненти

системи покликані максимально полегшити доступ споживачів до засобів ГІС, а сервіс має велику кількість налаштувань, а також можливість інтеграції створених карт у сайт.

Щоб користуватися послугою, у вас повинен бути обліковий запис Google, куди ви можете розміщувати ярлики та іншу просторову інформацію на карті. Вибираючи кольори та графіку, змінюється зовнішній вигляд міток та об'єктів, і графічні об'єкти можуть завантажуватися прямим посиланням на зображення. Зручність послуги дозволяє створювати мітки, за допомогою яких ви можете переглядати певні шари інформації на попередньо заповненій карті. Для міст, де доступна детальна інформація про окрему інфраструктуру, ви також можете переглянути «гібридну» карту, яка є просторовим зображенням із високою роздільною здатністю та всебічною топографією. Карти Google можна розміщувати на вдосконалених веб-сторінках, які безпосередньо посилаються на дані в базі даних Google. Це дозволяє централізувати та швидко оновлювати дані. Додаючи власні дані на карту, ви можете додавати власні об'єкти, маркери, інтерактивні підказки та спливаючі вікна. Дані для відображення можуть бути безпосередньо в коді веб-сторінки або у зовнішніх файлах. Система управління картою включає стандартні параметри навігації та анімації. Тематичні шари можна створювати безпосередньо у поданні веб-сторінки.

Особливістю Карт Google і його основною відмінністю є те, що дані зберігаються в Google, дані переглядаються та обробляються в Інтернеті. Переглядаючи дані на різних рівнях масштабування, досягається висока швидкість перегляду даних, і дані поступово перетягуються в міру завантаження. Дані організовані за принципом шару. Дані в системі готові до використання і не вимагають особливих заходів. Приватні дані є більш керованими, оскільки ви можете змінювати порядок їх появи (змінювати рядки), створювати рядки з декількома рядками та вбудовувати об'єкти. Будь-який користувач Google може переглядати створені дані. Робляться перспективні (рельєфні) зображення, малюються самі предмети, з'єднуються з поверхнею, вимірюється їх відстань, розглядається площа на певній висоті.

Для роботи з API спочатку потрібно зареєструватися, щоб отримати ключ API. Є кілька кроків, які можна зробити, щоб вставити картку на певну веб-сторінку. Розробник отримає унікальний код (ключ API), який дозволить вам користуватися сервісом Google Maps. Потім цей код слід знайти на сторінці карти та використати для ідентифікації користувача. Для підключення до послуги до тексту сторінки додаються лінії реєстрації, включаючи отриманий унікальний код. За допомогою простих команд ви можете додати на карту власні контури, маркери, інтерактивні поради та спливаючі вікна. Дані для відображення можуть бути безпосередньо в коді веб-сторінки або у зовнішніх файлах. API Maps є відкритою системою, тому розширення класів, створення майстра та користувальницьких пошукових систем забезпечує розширену взаємодію зі службою. Як і у випадку з іншими продуктами Google (GMail, Пошук Google, Документи Google тощо), служба поєднує потреби користувачів та можливості ГІС для взаємодії з просторовою інформацією.

Загальні засоби картографування ГІС служби картографування Google Maps можна використовувати майже по-різному, кожен із переваг та недоліків. Найповніші можливості надаються за допомогою інструментів програмування JavaScript із прямими викликами функцій API Google Maps. Спочатку потрібно створити HTML-код сторінки. Використовуючи API Карт Google, використовуючи документацію та інструкції щодо використання Карт Google. Потім створіть елемент на сторінці бібліотеки Google Maps API відбувається за допомогою скрипта на веб-сторінці (лістинг 3.1).

```
<script  
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCxsXXXX-  
kfjOFYyXXX&callback=initMap"></script>
```

Лістинг 3.1 Код виклику Google Maps API

Визначення об'єкту карти Google Maps, центрування карти параметром *center* та рівень наближення (масштабу) карти параметром *zoom*, а при необхідності – модифікування стилю відображення карти та її елементів за допомогою параметру *style* показано у лістинг 3.2.

```

map = new google.maps.Map(document.getElementById('map'), {
    center: {lat: 49.835199, lng: 24.0084904},
    zoom: 18
});

```

Лістинг 3.2 Код вибору параметрів відображення карти

Для відображення додаткової інформації ГІС, наприклад інформаційного та технологічного характеру на картах Google Maps доцільно застосовувати спеціальний елемент типу маркер. Для цього визначається змінна маркера (лістинг 3.3).

```

var marker = new google.maps.Marker({
    position: {lat: 49.835088, lng: 24.0085},
    map: map,
    title: 'Інформація по маркеру IoT1',
    icon: 'https://theXXX.com/redmarker48.png'
});

```

Лістинг 3.3 Код параметрів маркера

Додатковими параметрами маркера визначається його розташування *position*, назва *title* та піктограма *icon*. При підведенні курсору до маркера відобразатиметься вказана інформація про об'єкт (рисунок 3.3).

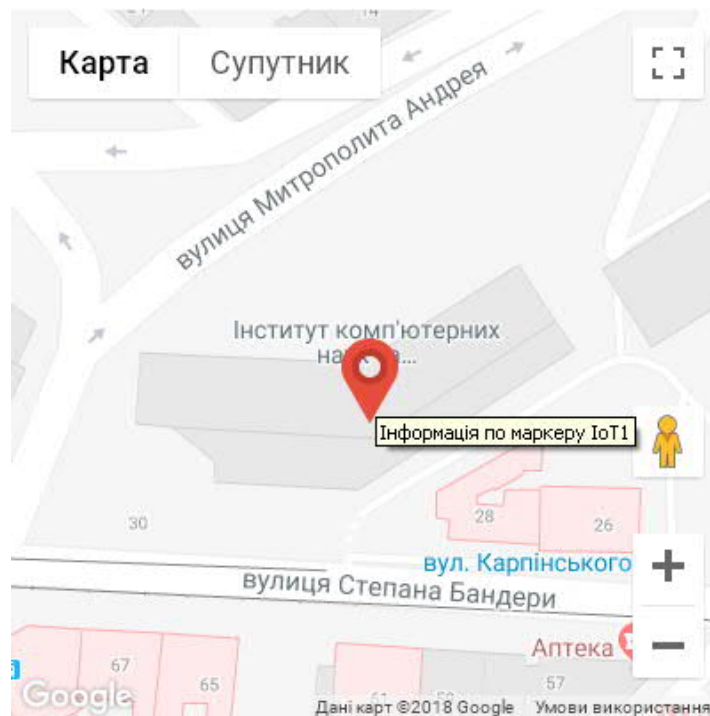


Рисунок 3.3 Отримане засобами ГІС Google Maps API зображення об'єкту на карті

Для відображення більш повної інформації про параметри, наприклад, технологічного об'єкту, який прив'язаний до маркера, доцільно використовувати окреме інформаційне вікно. Для цього слід додати подію при кліканні на маркер у Google Maps наступним чином (лістинг 3.4):

```
var infowindow = new google.maps.InfoWindow({
  content: '<p>Координати маркера:' + marker.getPosition()
          + '<br>Вимірюваний параметр:'
          + value_data + '</p>'
});
google.maps.event.addListener(marker, 'click', function() {
  infowindow.open(map, marker);
});
```

Лістинг 3.4. Код відпрацювання події

Вміст повідомлення інформаційного вікна описується у параметрі *content*. Перший параметр методу *event*, що описує реакцію на певну дію користувача, *addListener* включає ім'я відповідного об'єкту *marker*. Об'єктом може виступати маркер, або власне карта. Другий параметр *click* визначає тип події, в даному випадку – клікання мишкою. Третій параметр описує функцію-обробник, яка спрацює при настанні події (рисунок 3.4).

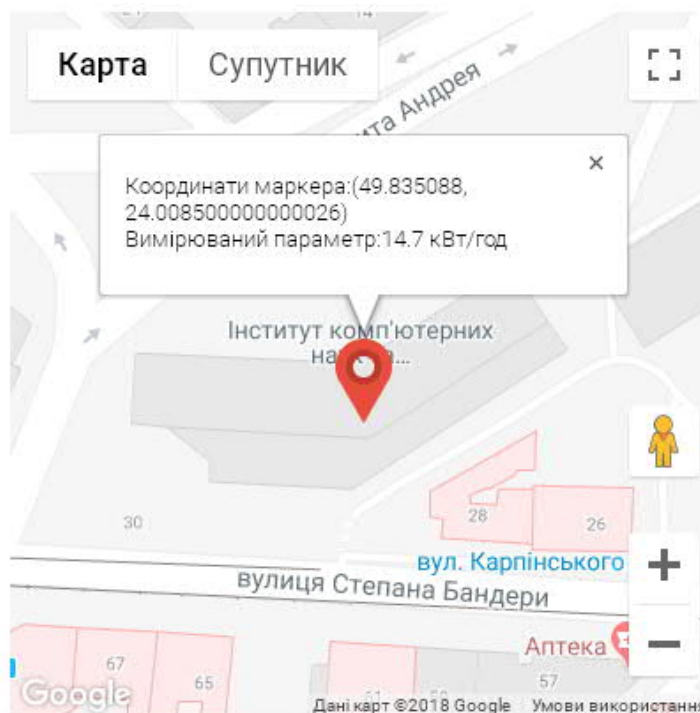


Рисунок 3.4 Відображення засобами Google Maps API параметрів технологічного об'єкта на карті

Цей підхід забезпечує найбільшу гнучкість у створенні користувацького програмного забезпечення для ГІС для потреб IASUE, але вимагає певних знань щодо ведення журналу та особливостей використання спеціального API Google Maps.

Інший метод полягає у використанні спеціальних веб-служб, які дозволяють створювати (генерувати) необхідну карту з даних користувачів. Однією з них є служба картографування BatchGeo, яка покликана значно спростити та пришвидшити процес картографування за допомогою даних користувачів для публікації в Інтернеті або друку. Додавання всіх точок на Карти Google вручну зазвичай займає багато часу, а також займе багато часу. Якщо користувач має велику кількість даних таблиці для перегляду в полі, рекомендується скористатися веб-службою BatchGeo. Це можуть бути різні технологічні деталі. Завдяки службі BatchGeo створюється інтерактивна карта, готова до сайту, без ручного кодування та передачі даних.

Далі йде процес створення карти з розташуванням та основними параметрами засобів збору інформації IASUE. Створіть таблицю даних для візуалізації за допомогою Excel, Google Docs або OpenOffice Calc (таблиця 3.2).

Таблиця 3.2 Параметрами технологічного об'єкту для створення карт

Name	Latitude	Longitude	Value	Type	Description
IoT 1 електроенергія	49.835199	24.0084904	123,1	1	кВт/г
IoT 2 вода	49.835088	24.0085	21,2	2	л/г
IoT 3 електроенергія	49.83501	24.0083	42,2	1	кВт/г
IoT 4 електроенергія	49.83503	24.0088	37,2	1	кВт/г
IoT 5 вода	49.83504	24.0086	92,1	2	л/г

У таблиці 3.2 важливо мати інформацію про координати об'єктів у полях, що називаються широтою та довготою.

На головну сторінку служби BatchGeo завантажується <https://batchgeo.com/> таблиця 1, яка описує наявні дані. Послуга підтримує імпорт даних з усіх основних редакторів електронних таблиць (Excel, Google Docs, OpenOffice Calc). Потім, слідуючи процесу геокодування, посилання на створену карту технологічного об'єкта буде надіслане на електронну пошту користувача послуги,



яке згодом може бути змінено за допомогою наданого посилання. На рисунку 3.5 показана карта, створена веб-службою BatchGeo, з відображеними параметрами процесу. Якщо потрібно, ви можете переглянути карту в службі Google Earth, вибравши опцію меню «Експортувати в Google Earth».

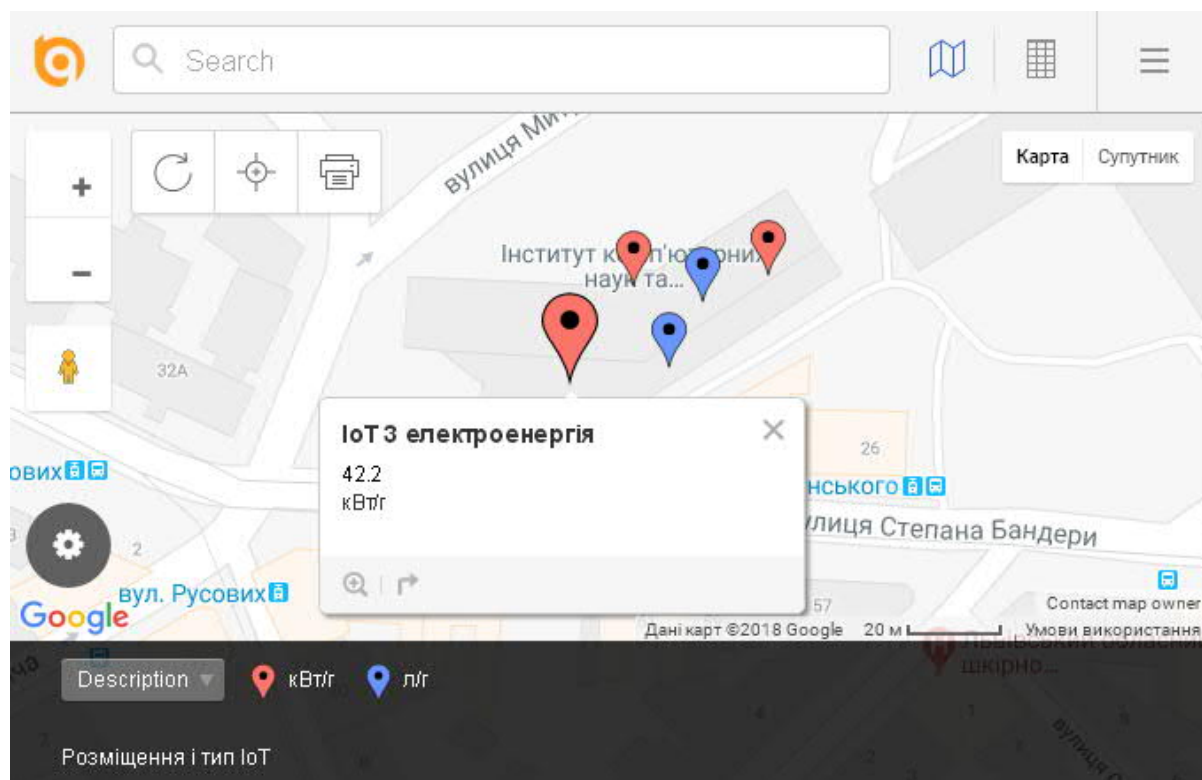


Рисунок 3.5 Карта з параметрами технологічного об'єкту створена засобами веб-сервісу BatchGeo

Однак найбільш гнучкі варіанти створення та розгортання інструментів відображення ГІС для IASUE забезпечуються використанням інтегрованих рішень Google Cloud. Через його використання для покращення та візуалізації даних не потрібно стороннє програмне забезпечення. Розгляньте можливість створення інтегрованого ГІС-інтерфейсу для взаємодії користувача з даними, що надсилаються низькорівневими компонентами за допомогою безкоштовної служби Google Cloud.

Після реєстрації свого облікового запису користувача на <http://www.google.com/accounts/>, вам потрібно увійти у свій обліковий запис (<https://drive.google.com/drive/my-drive>) і створити новий документ із: Кнопка "+

Новий" і виберіть "Мої карти Google". У вікні Служби карт ви можете вибрати тип папки, в якій відобразатимуться теги об'єктів та користувацькі дані, використовуючи опцію «Карта виводу». Далі вам потрібно додати новий шар ГІС, де реальні об'єкти, такі як IASUE, у нашому випадку відображаються за допомогою кнопки "Додати шар". Наприклад, цих рівнів може бути кілька залежно від типу об'єктів IASUE.

Потім додайте розташування окремого об'єкта, натиснувши кнопку «Додати маркер» і перетягнувши його до місця, що відповідає географічним координатам зазначеного об'єкта. Введіть ім'я об'єкта, параметри відображення, атрибути тощо у вікні властивостей. Вкажіть. І натисніть Зберегти. Так само вам потрібно додати його на карту та визначити всі необхідні об'єкти. Створена картка автоматично зберігається у вашому обліковому записі. Ви можете отримати посилання на карту, натиснувши кнопку "Огляд". Приклад розробленого таким чином ГІС-інтерфейсу наведено на рисунку 3.6.

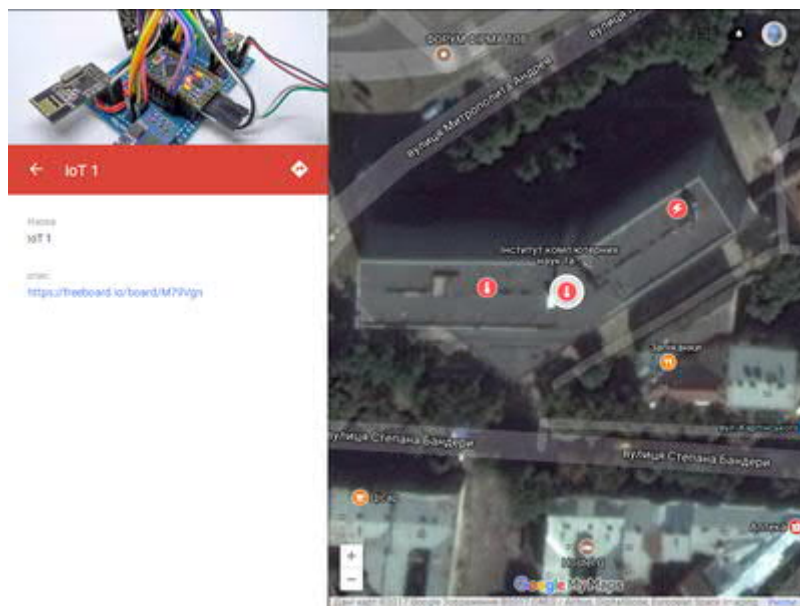


Рисунок 3.6 Інтерфейс ГІС

Створену карту можна опублікувати у мережі за допомогою сервісу веб-хостингу у Google Cloud (<https://sites.google.com/>). У вказаному сервісі після створення заготовки веб-сторінки кнопкою «+» слід у вікні вказати необхідні атрибути сторінки (назву, заголовки тощо) і через меню «Insert», «from Drive»

додати на сторінку створену карту. Кнопкою «Publish» відбувається автоматична публікація у мережі Інтернет та формується посилання на опубліковану веб-сторінку.

### 3.3 Висновки до розділу

Було реалізовано апарат нейронних мереж для збору та опрацювання даних про використання електроенергії. Розроблено засіб візуалізації та доступу до таких даних.

## ВИСНОВКИ

1. Рекомендується розробити IASUE регіональної економіки на основі компонентно-орієнтованої технології та інтегрованого підходу, що включає методи та засоби аналітичної обробки даних, моделювання, прогнозування та прийняття рішень.

2. Регіональна економіка IASUE була розроблена з метою створення єдиної галузі знань із надійною, всебічною та оперативною інформацією, яка використовується для прийняття ефективних управлінських рішень.

3. Інструменти збору даних були розроблені з використанням концепції Інтернету речей, заснованої на невеликих інтелектуальних датчиках, які розпорошені у просторі та підключаються до бази даних на хмарному сервері.

4. Було показано, що розробка компонентів геоінформаційної системи для регіональної економіки для IASUE повинна здійснюватися за допомогою хмарного сервісу Google Cloud Documents та спеціального API Google Maps, який дозволяє швидко створювати, змінювати та створювати необхідні компоненти, замінені . . . більші навички знань.

5. Додаткова участь засобів програмування, зокрема JavaScript із використанням API Google Maps, надає можливість розробити геоінформаційну систему для майбутньої економіки IASUE регіону, враховуючи додаткові специфічні потреби користувача системи. .

6. Спеціальні інструменти, доступні громадськості для перегляду та аналізу даних про енергоспоживання та енергоефективність економіки IASUE в регіоні, керівництво використанням регіону, баз даних та архівів даних, перевірка дійсності та забезпечення ефективності управління. рішення.

7. Візуалізація та обробка енергетичних даних виявились надзвичайно чутливими до людей з точним посиланням на розташування об'єктів управління.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Данило А.М., Серета Ю. І. Алгоритми та апаратна реалізація функцій сигмоїдальних функцій активації III Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020, Тернопіль, Україна. Тернопіль: ЗУНУ, 2020. с.62
2. Данило А.М., Серета Ю. І. Методи синтезу та створення нейромереж реального часу паралельно-вертикального типу III Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 26 листопада 2020, Тернопіль, Україна. Тернопіль: ЗУНУ, 2020. с.66.
3. Цмоць І.Г., Теслюк Т.В., Опотяк Ю.В. Теслюк В.М. Архітектура багаторівневої системи управління енергоефективністю регіону // Вісник НУ «Львівська політехніка» «Комп'ютерні науки та інформаційні технології», № 864. – Львів 2017. С. 201-209.
4. Цмоць І.Г., Скорохода О.В., Роман В.І. Сховища даних багаторівневих систем управління енергоефективністю // Збірник наукових праць «Моделювання та інформаційні технології». Інститут проблем моделювання в енергетиці. Випуск 77, 2016. С. 192-197.
5. І.Г. Цмоць, В.І. Роман. Вдосконалення методу групування енергетичних даних у системі багаторівневого управління енергоефективністю економіки регіону. Науковий вісник НЛТУ України, 2019. – т. 29, №1. – С.116 – 120.
6. Карпа Д.М., Цмоць І. Г., Теслюк В.М. Засоби підтримки прийняття рішень для визначення пріоритетності виконання енергозберігаючих проектів. Науковий вісник НЛТУ Українию, 2019. – т. 29, №2. – С.135 – 140.
7. Bertocco M., Cappellazzo S., Flammini A., Parvis M. A multi-layer architecture for distributed data acquisition // Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, 2002. IMTC/2002, vol. 2, 2002, pp. 1261-1264.

8. O’Leary, D.E. Enterprise resource planning systems: systems, life cycle, electronic commerce, and risk. Cambridge University Press, 2000.
9. Chiu, Yu Hsien, et al. Enterprise resource planning, 2014.
10. Data Warehouse Design [Электронный ресурс]. – Режим доступа: <http://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>
11. Big Data Implementation vs. Data Warehousing [Электронный ресурс]. – Режим доступа: <http://www.b-eye-network.com/view/17017>
12. Lambda Architecture [Электронный ресурс] – Режим доступа: <http://lambda-architecture.net/>
13. Big Data and its Impact on Data Warehousing [Электронный ресурс] – Режим доступа: [http://cdn.ttgtmedia.com/BeyeNETWORK/downloads/BigDataE-Book\\_final.pdf](http://cdn.ttgtmedia.com/BeyeNETWORK/downloads/BigDataE-Book_final.pdf)
14. Teslyuk T., Tsmots I., Teslyuk V., Medykovskyy M., Opotyak Y. Architecture and Models for System-Level Computer-Aided Design of the Management System of Energy Efficiency of Technological Processes at the Enterprise // Automation 2017. International Conference Automation 2017, Advances in Intelligent Systems and Computing 689, Springer, 2018. pp. 538-557.
15. Medykovskyy M.O., Tsmots I.G., Skorokhoda. O.V. Spectrum neural network filtration technology for improving the forecast accuracy of dynamic processes in economics // Actual Problems of Economics. №12(162) – 2014. pp. 410-416.
16. Medykovskyy M.O., Tsmots I.G., Skorokhoda. O.V. Teslyuk T.V. Design of Intelligent Component of Hierarchical Control System // Econtechmod. An International Quarterly Journal. – 2016. Vol. 5. No 2.3. pp. 3-10.
17. Medykovskyy M.O., Tsmots I.G., Tsymbal Y.V. Intelligent data processing tools in the systems of energy efficiency management for regional economy // Actual Problems of Economics. № 12(150) – 2013. pp. 271-277.
18. Tsmots I.G., Tsymbal Y.V., Tsmots O.I. Early-warning systems for enterprises using neural networks tools // Actual Problems of Economics. № 10 (136) – 2012. pp. 283-291.

19. Teslyuk T., Tsmots I., Teslyuk V., Medykovskiy M., Opotyak Y. Architecture of the management system of energy efficiency of technological processes at the enterprise // Proceedings of XIIth International Scientific and Technical Conference CSIT 2017. 5-8 September 2017. Lviv, Ukraine. pp. 429-433.

20. Medykovskiy M.O., Tsmots I.H. Tsymbal Y.V. Information analytical system for energy efficiency management at enterprises in the city of Lviv (Ukraine) // Actual Problems of Economics. №1 (175) – 2016. pp. 379-384.

21. Diamantaras K.I., Kung S.Y. Principal Component Neural Networks: Theory and Applications, Wiley, 1996. 270 p.

22. Tsymbal Y., Tkachenko R. A Digital Watermarking Scheme Based on Autoassociative Neural Networks of the Geometric Transformations Model // Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing. – 2016. – pp. 231-234.

23. Nikos Bikaks. "Big Data Visualization Tools" Encyclopedia of Big Data Technologies, Springer 2018. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/1801.08336.pdf>

24. Vitaly Friedman (2008) "Data Visualization and Infographics" in: Graphics, Monday Inspiration, January 14th, 2008 [Електронний ресурс]. – Режим доступу: <https://www.smashingmagazine.com/2008/01/monday-inspiration-data-visualization-and-infographics/>

25. "Data Visualization for Human Perception". The Interaction Design Foundation [Електронний ресурс]. – Режим доступу: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception>

26. Lengler, Ralph; Eppler, Martin. J. "Periodic Table of Visualization Methods" [Електронний ресурс]. – Режим доступу: [http://www.visual-literacy.org/periodic\\_table/periodic\\_table.html](http://www.visual-literacy.org/periodic_table/periodic_table.html)

27. Intellectual components of integrated automated control systems: monograph / Medykovsky M.O., Tkachenko R.O., Tsmots I.G., Tsymbal Y.V., Doroshenko A.V., Skorokhoda O.V. – Lviv: Lviv Polytechnic Publishing House, 2015. 280 p.

28. Medikovsky M.O., Tsmots I.G., Podolsky M.R. Substantiation of the principles of construction and development of the generalized structure of the information-analytical system for estimation, forecasting and management of energy efficiency of the region's economy // Bulletin of NU "Lviv Polytechnic" "Computer sciences and information technologies", № 751. – Lviv, 2013. C. 40- 51.

29. Tsmot I.G., Teslyuk T.V., Opotyak Yu.V., Teslyuk V.M. Architecture of a Multilevel Energy Efficiency Management System in the Region // Bulletin of the National Technical University of Lviv Polytechnic "Computer Science and Information Technologies", № 864. - Lviv 2017. P. 201-209.

30. Tsmots I.G., Skorokhoda O.V., Roman V.I. Data repositories of multilevel energy efficiency management systems // Collection of scientific papers "Modeling and information technologies". Institute of Modeling Problems in Energy. Issue 77, 2016. pp. 192-197.

31. I.G. Tsmots, V.I. Roman. Improving the method of grouping energy data in the system of multilevel energy efficiency management of the region's economy. Scientific Bulletin of the NLTU Ukraine, 2019. - Vol. 29, No. 1. - P.116 - 120.

32. Karpa D.M., Tsmots I. G., Teslyuk V.M. Decision support tools for prioritizing energy-saving projects. Scientific Bulletin of the NLTU Ukraine, 2019. - Vol. 29, No. 2. - P.135 - 140.

33. Bertocco M., Cappellazzo S., Flammini A., Parvis M. A multi-layer architecture for distributed data acquisition // Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, 2002. IMTC / 2002, vol. 2, 2002, pp. 1261-1264.

34. O'Leary, D.E. Enterprise resource planning systems: systems, life cycle, electronic commerce, and risk. Cambridge University Press, 2000.

35. Chiu, Yu Hsien, et al. Enterprise resource planning, 2014.

36. Data Warehouse Design. [Electronic resource] – Access Mode: <http://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>

37. Big Data Implementation Vs. Data Warehousing. [Electronic resource] – Access Mode: <http://www.b-eye-network.com/view/17017>



38. Lambda Architecture [Electronic resource] – Access mode: <http://lambda-architecture.net/>
39. Big Data and Its Impact on Data Warehousing [Online Resource] – Access Mode: [http://cdn.ttgtmedia.com/BeyeNETWORK/downloads/BigDataE-Book\\_final.pdf](http://cdn.ttgtmedia.com/BeyeNETWORK/downloads/BigDataE-Book_final.pdf)
40. Teslyuk T., Tsmots I., Teslyuk V., Medykovskyy M., Opotyak Y. Architecture and Models for System-Level Computer-Aided Design of the Energy Efficiency Management System of Technological Processes at the Enterprise // Automation 2017. International Conference Automation 2017, Advances in Intelligent Systems and Computing 689, Springer, 2018. pp. 538-557.
41. Medykovskyi M.O., Tsmots I.G., Skorokhoda. O.B. Spectrum neural network filtration technology for improving the forecast accuracy of dynamic processes in economics // Actual Problems of Economics. No. 12 (162) - 2014. pp. 410-416.
42. Medykovskyi M.O., Tsmots I.G., Skorokhoda. O.B. Teslyuk T.V. Design of Intelligent Component of Hierarchical Control System // Econtechmod. An International Quarterly Journal. - 2016. Vol. 5. No 2.3. pp. 3-10.
43. Medykovskyi M.O., Tsmots I.G., Tsymbal Y.V. Intelligent data processing tools in energy efficiency management systems for regional economy // Actual Problems of Economics. № 12 (150) - 2013. pr. 271-277.
44. Tsmots I.G., Tsymbal Y.V., Tsmots O.I. Early warning systems for enterprises using neural networks tools // Actual Problems of Economics. № 10 (136) - 2012. pr. 283-291.
45. Teslyuk T., Tsmots I., Teslyuk V., Medykovskyi M., Opotyak Y. Architecture of the management system of energy efficiency of technological processes in the enterprise // Proceedings of the XIIth International Scientific and Technical Conference CSIT 2017. 5- 8 September 2017. Lviv, Ukraine. pp. 429-433.
46. Medykovskyi M.O., Tsmots I.H. Tsymbal Y.V. Information analytical system for energy efficiency management at enterprises in the city of Lviv (Ukraine) // Actual Problems of Economics. №1 (175) - 2016. pp. 379-384.
47. Diamantaras K.I., Kung S.Y. Principal Component Neural Networks: Theory and Applications, Wiley, 1996. 270 p.

48. Tsymbal Y., Tkachenko R. A Digital Watermarking Scheme Based on Autoassociative Neural Networks of the Geometric Transformations Model // Proceedings of the 2016 IEEE First International Conference on Data Stream Mining & Processing. - 2016. - pp. 231-234.

49. Nikos Bulls. "Big Data Visualization Tools" Encyclopedia of Big Data Technologies, Springer 2018. [Electronic resource] – Access Mode: <https://arxiv.org/pdf/1801.08336.pdf>

50. Vitaly Friedman (2008) "Data Visualization and Infographics" in: Graphics, Monday Inspiration, January 14th, 2008. [Electronic resource] – Access Mode: <https://www.smashingmagazine.com/2008/01/monday-inspiration-data-visualization-and-infographics/>

51. "Data Visualization for Human Perception." The Interaction Design Foundation. Retrieved 2015-11-23. [Electronic resource] – Access Mode: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception>

52. Lengler, Ralph; Eppler, Martin. J. "Periodic Table of Visualization Methods". [www.visual-literacy.org](http://www.visual-literacy.org). [Electronic resource] – Access Mode: [http://www.visual-literacy.org/periodic\\_table/periodic\\_table.html](http://www.visual-literacy.org/periodic_table/periodic_table.html)