

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Розробка та дослідження методів для аналізу рухів людини на основі  
штучного інтелекту  
(тема)

Виконав:  
здобувач другого року навчання,  
групи СШМ-23-1

Тритяк Матвій  
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту  
(повна назва освітньої програми)

Керівник ас. Максим Єрохін  
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ \_\_\_\_\_  
(підпис)

Олег ЗОЛОТУХІН  
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Системи штучного інтелекту \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Третьяку Матвію Ігоровичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Розробка та дослідження методів для аналізу рухів людини на основі штучного інтелекту \_\_\_\_\_

затверджена наказом університету від 21 квітня 2025 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 6 червня 2025 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та наукових проєктів щодо розробки та дослідження глибоких нейронних мереж, набори даних для визначення активності людини, документація Python

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Аналіз предметної галузі та постановка задачі

2) Теоретичний огляд

3) Експериментальне дослідження

4) Результати проведеного дослідження



## РЕФЕРАТ

Пояснювальна записка: 90 с., 20 рис., 2 табл., 2 дод., 30 джерела.

БАГАТОЗАДАЧНЕ НАВЧАННЯ, ГЛИБИННІ НЕЙРОННІ МЕРЕЖІ, МЕРЕЖІ ПЕРЕСІЧНОГО ЗШИВАННЯ, ОБМІН ПАРАМЕТРАМИ, РОЗПІЗНАВАННЯ ДІЯЛЬНОСТІ ЛЮДИНИ.

Об'єкт дослідження – процес багатозадачного навчання з використанням глибинних нейронних мереж для розпізнавання діяльності людини в реальному часі.

Предмет дослідження – методологія побудови багатозадачних моделей на основі глибинних нейронних мереж, що включає застосування мереж пересічного зшивання та адаптивного обміну параметрами між підзадачами для розв'язання задач розпізнавання діяльності людини.

Мета дослідження – розробка та обґрунтування підходу до розпізнавання діяльності людини в реальному часі, що використовує багатозадачне навчання з елементами пересічного зшивання та обміну параметрами, що забезпечує підвищену точність та ефективність моделі.

Методи дослідження – методи глибинного навчання для побудови моделей розпізнавання, багатозадачне навчання для синхронізації підзадач, еволюційні алгоритми для налаштування архітектури нейронних мереж, методи математичної статистики для оцінки точності результатів, експериментальні дослідження ефективності застосування обміну параметрами між задачами.

Наукова новизна роботи – запропоновано новий підхід до розпізнавання діяльності людини, що поєднує багатозадачне навчання з глибинними нейронними мережами, методами пересічного зшивання та обміну параметрами, що дозволяє підвищити точність і адаптацію моделі до змінюваних умов у реальному часі.

## ABSTRACT

Master's thesis contains: 90 pp., 20 fig., 2 tabl., 2 ann., 30 references.

CROSS-STITCH NETWORKS, DEEP NEURAL NETWORKS, HUMAN ACTIVITY RECOGNITION, MULTI-TASK LEARNING, PARAMETER SHARING.

Object of the research – the process of multi-task learning using deep neural networks for real-time human activity recognition.

Subject of the research – the methodology of constructing multi-task models based on deep neural networks, including the use of cross-stitch networks and adaptive parameter sharing between subtasks for solving human activity recognition problems.

Research goal – the development and substantiation of an approach to real-time human activity recognition that utilizes multi-task learning with elements of cross-stitching and parameter sharing, ensuring increased accuracy and model efficiency.

Research methods – deep learning methods for building recognition models, multi-task learning for synchronizing subtasks, evolutionary algorithms for tuning neural network architectures, mathematical statistics for evaluating accuracy, and experimental studies of the effectiveness of parameter sharing between tasks.

Scientific novelty of the work – a new approach is proposed for human activity recognition that combines multi-task learning with deep neural networks, cross-stitching methods, and parameter sharing, enabling improved accuracy and adaptability of the model to changing conditions in real time.

## ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі та постановка задачі .....	10
1.1 Опис предметної галузі.....	10
1.2 Актуальність дослідження.....	11
1.3 Постановка задачі .....	13
2 Теоретичний огляд .....	14
2.1 Глибинні нейронні мережі.....	14
2.1.1 Багатошаровий персептрон .....	16
2.1.2 Згорткові нейронні мережі .....	19
2.1.3 Рекурентні нейронні мережі.....	21
2.2 Багатозадачне навчання .....	24
2.2.1 Багатозадачне навчання у нейронних мережах.....	25
2.2.2 Переваги багатозадачного навчання .....	28
2.2.3 Попередні дослідження в сфері багатозадачного навчання.....	29
3 Експериментальне дослідження.....	31
3.1 Основні типи архітектур та їх особливості.....	31
3.1.1 Однозадачне навчання .....	31
3.1.2 Жорсткий обмін параметрами.....	32
3.1.3 М'який обмін параметрами .....	33
3.1.4 Мережі для пересічного навчання .....	35
3.2 Використані набори даних для дослідження .....	37
3.2.1 Набору даних Opportunity.....	37
3.2.2 Опис набору даних Deep.....	40
3.3 Вибір архітектури для дослідження .....	44
3.3.1 Опис архітектури для набору даних Opportunity .....	44
3.3.2 Опис архітектури для набору даних Deep .....	47
3.4 Характеристика та особливості технічної реалізації .....	47
3.5 Представлення розроблених архітектур.....	49

4 Результати проведеного дослідження .....	55
4.1 Опис процесу оцінки моделі .....	55
4.2 Жорсткий обмін параметрами .....	56
4.2.1 Набір даних Opportunity.....	56
4.2.2 Набір даних Dear.....	58
4.2.3 Порівняльний аналіз жорсткого обміну параметрів для розглянутих наборів даних .....	59
4.3 М'який обмін параметрами .....	61
4.3.1 Набір даних Opportunity.....	61
4.3.2 Набір даних Dear.....	62
4.3.3 Порівняльний аналіз м'якого обміну параметрів для розглянутих наборів даних .....	63
4.4 Мережа пересічного зшивання.....	64
4.4.1 Набір даних Opportunity.....	64
4.4.2 Набір даних Dear.....	66
4.4.3 Порівняльний аналіз мережі пересічного зшивання параметрів для розглянутих наборів даних .....	67
4.5 Порівняльний аналіз розглянутих підходів .....	68
Висновки.....	71
Перелік джерел посилання .....	73
Додаток А Вигляд створених архітектур .....	77
Додаток Б Програмний код .....	81
Додаток В Відомість кваліфікаційної роботи.....	90

## ВСТУП

Глибинні нейронні мережі (ГНМ) є одним із найефективніших методів машинного навчання, що дозволяє аналізувати складні структури даних, зокрема часові ряди, які характеризують рухову активність людини. Отримання таких даних можливе завдяки сенсорним пристроям, серед яких акселерометри, гіроскопи, магнітометри та системи оптичного захоплення руху. Аналіз рухів людини має широке застосування в медицині, реабілітації, спортивній аналітиці, системах безпеки та інтелектуальних пристроях.

Однак використання ГНМ у цій галузі супроводжується низкою викликів, серед яких особливу проблему становить потреба у великих обсягах високоякісних анотованих даних, необхідних для ефективного навчання моделей, а також їхня недостатня стійкість до варіативності умов. Традиційні підходи не завжди забезпечують достатню точність і здатність до узагальнення, що обмежує можливості практичного застосування нейронних мереж для розпізнавання рухових патернів.

Одним із перспективних напрямів розв'язання цієї проблеми є багатозадачне навчання (Multi-Task Learning, MTL), що дозволяє моделі одночасно навчатися на кількох взаємопов'язаних задачах, що сприяє покращенню її продуктивності та зменшенню вимог до навчальних вибірок [9]. Дослідження методів MTL у контексті аналізу рухової активності людини є актуальним завданням, оскільки їхнє застосування дозволяє підвищити ефективність класифікації рухів, оптимізувати використання даних та забезпечити адаптивність моделей у різних сценаріях.

У зв'язку з цим об'єктом дослідження є процеси розпізнавання, аналізу та класифікації рухової активності людини на основі даних сенсорних систем. Предметом дослідження є методи багатозадачного

навчання у глибоких нейронних мережах та їх використання для аналізу часових рядів, що відображають рухові патерни людини.

Метою дослідження є оцінка ефективності багатозадачного навчання у задачах розпізнавання та класифікації рухів людини, визначення оптимальних архітектур нейронних мереж для обробки сенсорних даних і розробка рекомендацій щодо практичного застосування MTL у галузі аналізу людської моторики.

Наукова новизна дослідження полягає у проведенні систематичного аналізу впливу багатозадачного навчання на точність і стійкість моделей у задачах розпізнавання рухових патернів, дослідженні ефективності спільного навчання взаємопов'язаних завдань, розробці рекомендацій щодо вибору архітектур глибоких нейронних мереж, адаптованих до специфіки сенсорних даних, а також обґрунтуванні перспектив застосування багатозадачного навчання для підвищення стійкості моделей до варіативності рухів та обмеженості навчальних вибірок.

Результати дослідження сприятимуть подальшому розвитку методів аналізу людської моторики та вдосконаленню підходів до розпізнавання рухової активності у різних практичних застосуваннях.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Опис предметної галузі

Стрімкий розвиток глибинних нейронних мереж зумовив появу нових підходів до їхньої оптимізації, що стало можливим завдяки значному прогресу в апаратному забезпеченні та алгоритмічних удосконаленнях. Серед таких підходів особливу увагу привертає багатозадачне навчання, яке, хоча й було запропоноване ще наприкінці ХХ століття, набуло широкого застосування лише в останні роки. Основна ідея цього методу полягає у навчанні однієї моделі виконанню кількох взаємопов'язаних завдань, що дозволяє ефективніше використовувати наявні дані та покращувати узагальнюючу здатність моделей. У контексті аналізу рухової активності людини та розпізнавання її моторних патернів застосування багатозадачного навчання відкриває нові можливості для підвищення точності класифікації, адаптації моделей до різноманітних умов та зменшення вимог до розміру навчальних вибірок [1].

Розвиток сенсорних технологій, зокрема акселерометрів, гіроскопів, електроміографічних датчиків та систем оптичного відстеження рухів, сприяє накопиченню великих обсягів часових рядів, що містять інформацію про рухову активність людини. Однак їхня обробка традиційними методами стикається з низкою проблем, серед яких варіативність рухових патернів, наявність шумів, неоднорідність даних та необхідність адаптації моделей до нових умов. Використання багатозадачного навчання у цьому контексті дає змогу одночасно розв'язувати декілька споріднених завдань, таких як класифікація рухів, оцінка їхньої тривалості, прогнозування наступних дій та ідентифікація індивідуальних особливостей моторики. Це дозволяє покращити якість розпізнавання та підвищити стійкість моделей до змінюваних умов, що є критично важливим для застосувань у медицині, спорті, системах безпеки та інтерактивних технологіях.

З когнітивної точки зору багатозадачне навчання відповідає природним механізмам засвоєння інформації людиною, оскільки нові знання найкраще засвоюються у взаємозв'язку з уже наявною інформацією. Аналогічно, нейронні мережі, що навчаються одночасно виконанню кількох споріднених завдань, отримують змогу переносити знання між ними, що сприяє покращенню їхньої ефективності та узагальнюючої здатності. Такий підхід дозволяє значно зменшити залежність від великих обсягів анотованих даних, що є однією з ключових проблем у задачах розпізнавання рухової активності людини.

Таким чином, сучасний розвиток багатозадачного навчання відкриває нові перспективи у використанні глибинних нейронних мереж для аналізу рухів людини. Оптимізація архітектур моделей та ефективне використання взаємопов'язаних завдань дають змогу покращити точність розпізнавання, адаптивність до індивідуальних особливостей користувачів та стійкість до зміни умов. Це робить багатозадачне навчання перспективним напрямом досліджень, який може сприяти розвитку нових методів аналізу моторної активності та розширенню можливостей автоматизованих систем моніторингу та підтримки здоров'я.

## 1.2 Актуальність дослідження

Актуальність дослідження обумовлена стрімким розвитком технологій, що використовують глибинні нейронні мережі для аналізу людської активності. В останні роки спостерігається значний прогрес у застосуванні методів багатозадачного навчання (MTL), яке здатне одночасно вирішувати кілька взаємопов'язаних завдань, що зумовлює підвищення ефективності розпізнавання та класифікації різних типів рухової активності [2]. Важливість цього підходу полягає в тому, що багатозадачне навчання дозволяє досягати кращих результатів порівняно з традиційними методами, коли система навчається лише на одному завданні.

Зокрема, завдяки спільному використанню знань, отриманих від різних завдань, моделі набувають здатності краще узагальнювати результати, що робить їх більш адаптивними до нових, раніше не побачених даних.

Технології, що базуються на аналізі рухів людини за допомогою сенсорних пристроїв, таких як акселерометри, гіроскопи та інші біометричні датчики, стали невід'ємною частиною різних сфер, від медицини та спорту до систем безпеки та розважальних технологій. У медицині вони використовуються для моніторингу пацієнтів, оцінки фізичного стану, діагностики неврологічних порушень або виявлення падінь. В спортивній галузі ці методи дозволяють аналізувати рухову активність спортсменів, що забезпечує поліпшення результатів тренувань та запобігання травмам. У технологіях безпеки та робототехніці застосування таких систем дозволяє створювати адаптивні інтерфейси для людей з обмеженими можливостями, а також використовувати їх у автоматизованих системах моніторингу для покращення рівня безпеки. У розважальній індустрії такі технології дозволяють створювати реалістичні анімації та інтерактивні віртуальні персонажі, що особливо важливо для ігор та кінематографу.

З іншого боку, впровадження багатозадачного навчання в системи розпізнавання людської активності стикається з рядом серйозних викликів. Одним з основних є необхідність в обробці великих обсягів даних, які часто є нерівномірними або неповними, що вимагає розробки нових методів, здатних забезпечити стійкість і точність систем при роботі з такими вибірками [4]. Крім того, важливою задачею є адаптація моделей до різних умов і сценаріїв, що вимагає постійного удосконалення алгоритмів та архітектур нейронних мереж. У цьому контексті методи жорсткого та м'якого поширення параметрів, які є одними з найбільш вивчених, залишаються важливими інструментами, але існують і нові підходи, зокрема метод перехресного зшивання, які демонструють обнадійливі результати в задачах аналізу рухової активності.

Необхідність подальшого вдосконалення методів багатозадачного навчання та їх адаптація до нових вимог розпізнавання людської активності є надзвичайно актуальним завданням. Ці технології не тільки сприяють покращенню результатів в ряді галузей, але й відкривають нові можливості для застосування в таких сферах, як персоналізоване здоров'я, спорту, а також в індустрії безпеки та розваг. Розробка нових методів і вдосконалення існуючих підходів дозволяє не лише покращити точність класифікації, але й знизити вимоги до обсягу навчальних даних, що робить ці технології більш доступними і ефективними в практичному використанні.

### 1.3 Постановка задачі

Враховуючи важливість цього напрямку для широкого спектра застосувань, таких як медичний моніторинг, безпека та розваги, необхідно здійснити такі етапи для досягнення поставленої мети:

- провести аналіз основних методів багатозадачного навчання, їх особливостей та переваг у контексті класифікації людської активності;
- виконати порівняння методів жорсткого і м'якого обміну параметрами з новим підходом мереж пересічного зшивання;
- оцінити та вибрати оптимальні архітектури нейронних мереж для класифікації рухів людини, зокрема з використанням даних з датасетів Opportunity та Deap;
- розробити методику оцінки ефективності кожного підходу, враховуючи точність, швидкість навчання та здатність до узагальнення;
- реалізувати вибрані архітектури нейронних мереж та провести їх тестування на зазначених датасетах;
- проаналізувати отримані результати та порівняти ефективність різних підходів з урахуванням специфіки кожного датасету;
- розробити рекомендації щодо вибору оптимального методу для класифікації активності в залежності від умов застосування.

## 2 ТЕОРЕТИЧНИЙ ОГЛЯД

### 2.1 Глибинні нейронні мережі

Штучні нейронні мережі (ШНМ) не є новим феноменом, однак їх впровадження в реальні задачі довгий час обмежувалося розвитком технологій, які не забезпечували необхідної ефективності порівняно з іншими підходами. Але з розвитком обчислювальних потужностей та апаратного й програмного забезпечення, стало можливим використання не лише базових ШНМ, а й значно складніших і більш потужних моделей – глибинних нейронних мереж (ГНМ). Хоча терміни ШНМ та ГНМ часто використовуються як синоніми, в даному дослідженні доцільно вживати саме термін «глибинні нейронні мережі», що визначаються як мережі, що містять не менше ніж два приховані шари. Основна ідея таких мереж аналогічна роботі нейронів у біологічних системах, хоча сучасні нейронні мережі не є точними моделями функціонування мозку, а скоріше є спрощеними наближеннями [4].

Глибинні нейронні мережі складаються з численних зв'язаних нейронів, які обробляють інформацію в кількох шарах. Кожен нейрон здійснює перетворення вхідних даних через певну функцію активації, що визначає його вихід. Вихід мережі представлений у вигляді вектора, розмірність якого залежить від конкретної задачі. Наприклад, при класифікації зображень або тексту, вектор може містити ймовірності для різних класів, що відображають впевненість моделі в своєму прогнозі.

Процес створення нейронної мережі передбачає кілька етапів. Спочатку визначається її архітектура, що вибирається на основі завдання, яке необхідно вирішити. Для задач, пов'язаних з обробкою зображень, зазвичай використовуються згорткові нейронні мережі, а для роботи з текстовими даними чи послідовностями – рекурентні нейронні мережі.

Вибір архітектури також залежить від типу та обсягу даних, наявних обчислювальних ресурсів і вимог до швидкості обробки.

Після визначення архітектури, наступним етапом є навчання мережі, основною метою якого є мінімізація помилки прогнозу. Це досягається шляхом налаштування параметрів мережі, де для оцінки якості прогнозу використовують функцію втрат, що вимірює різницю між фактичними й прогнозованими результатами. В основі навчання лежить алгоритм градієнтного спуску, який спрямований на мінімізацію функції втрат шляхом поступових змін параметрів мережі. Зокрема, стохастичний градієнтний спуск і вдосконалені методи, такі як метод моментуму, адаптивний градієнтний спуск і алгоритм Adam, є популярними варіантами для оптимізації навчання.

Процес навчання з учителем передбачає подачу навчальних даних, що мають відповідні мітки, і мережа вчиться знаходити зв'язок між вхідними даними і цими мітками. Це дозволяє моделі в подальшому класифікувати невідомі дані. Мінімізація помилки між прогнозом і фактичними мітками є основною метою навчання.

Процес навчання (мінімізації функції втрат) можна формулювати наступним чином [5]:

$$\min_{\Theta \in R^n} L(\hat{Y}, Y) = \min_{\Theta \in R^n} L(F(X|\Theta), Y), \quad (2.1)$$

де  $\hat{Y}$  – прогнозовані значення виходів мережі;

$Y$  – істинні значення (мітки);

$X$  – вхідні дані;

$\Theta$  – параметри мережі.

Для обчислення градієнтів і функції помилки за допомогою міні-партій, алгоритм використовує кожну партію для обчислення градієнта, після чого оновлює параметри за правилом:

$$\Theta_{t+1} = \Theta_t - \eta \cdot \nabla_{\Theta} L, \quad (2.2)$$

де  $\Theta_t$  – параметри на поточному кроці  $t$ ;

$\eta$  – швидкість навчання (гіперпараметр);

$\nabla_{\Theta} L$  – градієнт функції втрат за параметрами мережі.

Таким чином, дані формули визначають процес навчання, який полягає в поступовому оновленні параметрів мережі для мінімізації різниці між прогнозованими та істинними значеннями на кожному кроці навчання.

### 2.1.1 Багатошаровий перцептрон

Багатошарові перцептрони (MLP) є основним типом глибоких нейронних мереж, який широко застосовується для вирішення завдань класифікації, регресії та інших типів машинного навчання. Вони складаються з кількох шарів, де кожен нейрон попереднього шару з'єднаний з кожним нейроном наступного шару, що дозволяє мережі навчатися складним патернам у даних [3].

Розглянемо математичну модель перцептрон, який складається з входу  $x \in R^n$ , матриці ваг  $W \in R^{n \times m}$ , коефіцієнта зміщення  $b \in R$  та виходу  $y \in R$ . Математичний вираз для активації нейрону виглядає так:

$$y(x|W, b) = \varphi(x \cdot W + b), \quad (2.3)$$

де  $x \cdot W$  – комбінацій вхідних значень  $x$ , зважених на ваги матриці  $W$ ;

$\varphi(\cdot)$  – нелінійна функція активації, що дозволяє мережі моделювати зсув для вхідних даних.

Функція активації  $\varphi$  є ключовим елементом моделювання нелінійних зв'язків між вхідними даними та виходами мережі. Вибір конкретної функції

активації залежить від задачі та її вимог. До найбільш поширених функцій активації належать:

– sigmoid. Використовується для задач бінарної класифікації, перетворює значення у діапазон  $(0, 1)$ , що інтерпретується як ймовірність;

– tanh. Гіперболічний тангенс, який перетворює значення у діапазон  $(-1, 1)$ , корисний для задач, де потрібно мати більшу чутливість до від'ємних значень;

– relu (Rectified Linear Unit). Функція, яка відсікає від'ємні значення, залишаючи тільки додатні, що сприяє швидшому навчанню та ефективному вирішенню проблеми зникаючих градієнтів;

– softmax. Зазвичай використовується на вихідному шарі для задач багатокласової класифікації, де потрібно перетворити вектори вихідних значень у ймовірності для кожного класу.

Типова архітектура MLP складається з трьох основних типів шарів: вхідного, прихованого та вихідного. У кожному шарі нейрони з'єднані з нейронами попереднього шару:

а) вхідний шар. Цей шар не здійснює обчислень. Він просто приймає вхідні дані (наприклад, ознаки зображення або тексту) і передає їх до наступного шару. Вхідний шар є обов'язковим, оскільки забезпечує зв'язок між вхідними даними та мережею;

б) приховані шари. Ці шари виконують основну роботу з навчанням, оскільки вони відповідальні за створення ознак та навчання нелінійних залежностей між ними. Кількість прихованих шарів і нейронів у кожному шарі зазвичай є гіперпараметрами, що визначаються в процесі налаштування моделі. Приховані шари здійснюють перетворення вхідних даних за допомогою лінійних та нелінійних операцій, навчаючись зворотним поширенням помилки;

в) вихідний шар. Цей шар генерує фінальний результат роботи мережі. Кількість нейронів на виході залежить від специфіки задачі:

– для задачі двокласової класифікації використовують один нейрон, який дає результат у вигляді ймовірності належності до одного з двох класів;

– для багатокласової класифікації кількість вихідних нейронів дорівнює кількості класів, і кожен нейрон видає ймовірність належності до певного класу;

– для задач регресії кількість вихідних нейронів може бути однаковою з кількістю цільових змінних або іншою залежно від специфіки задачі.

У задачах класифікації, зокрема багатокласовій класифікації, нейрони вихідного шару зазвичай подають свої значення до функції Softmax [30]. Ця функція перетворює скалярні виходи в ймовірності для кожного класу. Процес виглядає так:

$$P(y_i = 1|x) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \quad (2.4)$$

де  $z_i$  – значення, отримане від нейрона  $i$  на вихідному шарі;

$K$  – кількість класів;

$e^{z_i}$  – експоненціальна функція для кожного значення  $z_i$ .

Функція Softmax гарантує, що всі ймовірності будуть в діапазоні  $(0, 1)$  і сума ймовірностей для всіх класів буде дорівнювати одиниці. Для задач класифікації вибір вихідного класу здійснюється на основі найбільшої ймовірності.

Навчання багат шарового перцептрону здійснюється за допомогою алгоритму зворотного поширення помилки (backpropagation) [18]. Основний процес навчання включає:

– пряме поширення. Вхідні дані пропускаються через мережу, і кожен нейрон у кожному шарі обчислює своє значення відповідно до відповідної функції активації;

– обчислення помилки. На виході мережі обчислюється різниця між передбаченням мережі та реальними мітками класів або значеннями, за допомогою функції втрат (наприклад, крос-ентропії для задач класифікації);

– зворотне поширення помилки. Помилка з вихідного шару поширюється назад через мережу, і на основі цієї помилки обчислюються градієнти ваг. Ці градієнти використовуються для оновлення ваг за допомогою алгоритму оптимізації, такого як градієнтний спуск.

Отже, багат шарові персептрони – потужний інструмент для вирішення широкого спектру задач. Їх здатність автоматично виявляти складні залежності у даних та навчатися з використанням алгоритмів зворотного поширення помилки робить їх основним елементом у багатьох сучасних системах машинного навчання та штучного інтелекту.

### 2.1.2 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) є потужним інструментом для вирішення задач обробки зображень, оскільки вони дозволяють автоматично виділяти корисні ознаки з вхідних даних, що мають структуровану природу, як у випадку з зображеннями [30]. Головною операцією в CNN є операція згортки, що полягає в застосуванні фільтра (ядра згортки) до зображення або іншого вхідного тензора для виділення характеристик, таких як контури, текстури чи кольорові переходи.

Процес згортки можна описати математично за допомогою операції, яка обчислює скалярний добуток між фільтром  $K$  та вхідним зображенням  $I$ . Формула для операції згортки може бути записана наступним чином [14]:

$$S(i, j) = (K \cdot I)(i, j) = \sum_m \sum_n K(m, n) I(i + m, j + n), \quad (2.5)$$

де  $S(i, j)$  – значення, що виникає в точці  $(i, j)$  результату згортки;

$K$  – фільтр або ядро згортки, яке рухається по зображенню;

$I$  – вхідне зображення чи активаційна карта.

$(i, j)$  – індекси результату згортки, що відповідають позиції вхідного тензора;

$m, n$  – індекси, що позначають елементи фільтра.

У результаті виконання цієї операції на кожній позиції  $(i, j)$  вихідної матриці утворюється нове значення, яке є результатом комбінування локальних даних зображення з елементами фільтра. Операція згортки дозволяє мережі автоматично виявляти патерни на різних рівнях, починаючи з простих геометричних форм (наприклад, ліній) і закінчуючи складними текстурами та об'єктами [6].

Стандартна архітектура CNN складається з кількох основних елементів:

- згорткові шари (Convolutional Layers). На цих шарах виконуються операції згортки, де кожен фільтр відповідає за виявлення певних ознак;

- шари пулінгу (Pooling Layers). Ці шари зменшують розміри активаційних карт, зберігаючи лише найважливіші ознаки. Вони сприяють зниженню обчислювальної складності та зменшенню ймовірності перенавчання;

- повністю з'єднані шари (Fully Connected Layers). На цих шарах відбувається класифікація на основі виділених ознак з попередніх шарів.

Операція згортки дозволяє мережам виявляти локальні патерни, що є основним фактором ефективності CNN у задачах зображень. Крім того, використання пулінгу допомагає зменшити розміри даних, що надходять у наступні шари, тим самим зменшуючи кількість параметрів і обсяг обчислень.

У свою чергу, шари пулінгу виконують функцію зменшення розмірності активаційних карт. Зазвичай для цього використовуються такі типи пулінгу:

– максимальний пулінг (Max Pooling). Вибирається максимальне значення в кожному підвікні;

– середній пулінг (Average Pooling). обчислюється середнє значення елементів в кожному підвікні.

Розглянемо формулу для операції максимального пулінгу, де кожен елемент результату є максимумом з відповідного блоку вхідної активаційної карти [7]:

$$P(i, j) = \max(I(i + m, j + n)). \quad (2.6)$$

Кінцевий результат роботи CNN – це класифікація або регресія, в залежності від задачі. Для задач класифікації часто використовується останній шар, який може бути представлений як шар softmax для багатокласової класифікації. Цей шар перетворює отримані скалярні значення на ймовірності для кожного класу, що дає змогу визначити, до якого класу належить оброблене зображення.

Загалом, завдяки своїй здатності автоматично виділяти корисні ознаки, зменшувати розміри даних і виконувати ефективну класифікацію, CNN стали стандартом для задач, пов'язаних з обробкою зображень, а також відео та інших складних структурованих даних.

### 2.1.3 Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) є особливим класом нейронних мереж, що використовуються для обробки послідовних даних. Вони мають здатність зберігати інформацію про попередні етапи обробки, що робить їх ідеальними для задач, де важливий контекст або залежності між елементами

вхідної послідовності, наприклад, у задачах обробки природних мов, прогнозування часових рядів або обробки відео [3].

Основна особливість рекурентних нейронних мереж полягає в тому, що кожен нейрон у мережі має зв'язки не лише з нейронами попереднього шару, а й з самим собою на наступному етапі. Це дозволяє моделі «пам'ятати» інформацію з попередніх кроків, що є важливим для аналізу та прогнозування даних, де поточний стан залежить від попередніх.

Рекурентний нейронний елемент можна описати через рекурентну формулу, яка визначає стан мережі в поточний момент часу  $t$ :

$$h_t = \sigma(W_z x_t + W_h h_{t-1} + b), \quad (2.7)$$

де  $h_t$  – вектор стану мережі в момент часу  $t$ ;

$x_t$  – вхідний вектор на кроці  $t$ ;

$h_{t-1}$  – вектор стану мережі на попередньому кроці  $t-1$ ;

$W_x, W_z$  – вагові матриці для вхідних даних та попереднього стану відповідно;

$b$  – вектор зміщення;

$\sigma$  – активаційна функція.

Ця формула відображає те, як мережа комбінує поточні вхідні дані з попереднім станом для генерації нового стану, що буде передано на наступний крок.

Архітектура рекурентних нейронних мереж складається з декількох основних компонентів:

– вхідний шар. Приймає послідовні вхідні дані (наприклад, слова в тексті чи пікселі в відео);

– рекурентний шар. Основний елемент мережі, що здійснює рекурентну обробку даних. Кожен елемент цього шару залежить від

поточних вхідних даних та попереднього стану, що дозволяє мережі моделювати залежності в послідовностях;

– вихідний шар. На основі останнього стану або всіх попередніх станів, мережа генерує прогноз або класифікацію.

Зазвичай рекурентні мережі використовуються для задач, де важливо враховувати часову залежність або послідовність. Це включає такі задачі, як:

- прогнозування часових рядів;
- переклад тексту;
- розпізнавання мови;
- обробка природних мов (NLP).

У класичних RNN є проблема, з якою зіштовхуються при роботі з довгими послідовностями – втрата градієнта при зворотньому поширенні помилки. Це відбувається через те, що під час зворотного поширення помилки градієнти можуть зменшуватися або збільшуватися занадто сильно, що робить навчання мережі неефективним.

Для вирішення цієї проблеми були розроблені спеціалізовані архітектури, такі як LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit) [26]. Вони включають додаткові механізми, які дозволяють зберігати важливу інформацію про довгострокові залежності в послідовностях.

Після обробки послідовності в рекурентних мережах результати з усіх або лише з останнього стану можуть бути передані на вихідний шар для класифікації або прогнозування. Наприклад, у задачах класифікації кожен вихід мережі відповідає ймовірності належності до певного класу. Для цього найчастіше використовується функція активації softmax, яка перетворює активаційні значення в ймовірності.

Рекурентні нейронні мережі володіють рядом переваг, зокрема:

- можливість моделювати часові залежності у даних;
- здатність працювати з послідовними даними будь-якої довжини.

Однак вони також мають деякі обмеження, зокрема через труднощі в навчанні на довгих послідовностях (проблеми з градієнтом) і високі вимоги до обчислювальних ресурсів, що стимулювало розвиток інших архітектур, таких як LSTM або GRU.

Рекурентні нейронні мережі є важливим інструментом для роботи з послідовними даними. Вони мають широке застосування в різноманітних сферах, включаючи обробку природних мов, прогнозування часових рядів, і навіть у задачах комп'ютерного зору. Завдяки своїй здатності зберігати інформацію про попередні етапи обробки, ці мережі здатні досягати високих результатів у задачах, де важливі контекстуальні або тимчасові залежності.

## 2.2 Багатозадачне навчання

Багатозадачне навчання (БЗН, MTL) є підходом в галузі машинного навчання, який дозволяє моделі навчатися одночасно для кількох завдань, що можуть бути взаємопов'язаними [5]. Ключовою ідеєю цього методу є можливість спільного навчання, при якому модель оптимізується для розв'язання не лише основного завдання, а й декількох допоміжних завдань. Це дозволяє покращити загальну ефективність моделі, оскільки навчання на кількох задачах одночасно дозволяє використовувати спільні характеристики та залежності між ними.

Однією з важливих переваг БЗН є те, що модель не обмежується лише одним завданням, а використовує інформацію з інших задач, що можуть бути близькими за контекстом або структурою. В результаті, модель може краще узагальнювати знання, які вона отримує, та застосовувати їх до різних, але споріднених завдань. Це особливо важливо для задач, де дані обмежені, адже багатозадачне навчання дозволяє використовувати інформацію з різних джерел для поліпшення загальних результатів.

Завдання в багатозадачному навчанні можна умовно поділити на два основних типи:

– основні завдання, для яких продуктивність має таку ж важливість, як і для основного завдання. У такому випадку, усі задачі розглядаються як рівноцінні, і результат на кожному з них має значення для загальної ефективності моделі;

– додаткові завдання, що використовуються для поліпшення результатів основного завдання. Продуктивність на таких завданнях не є критичною, але вони сприяють кращому навчанні основної задачі шляхом надання додаткових контекстних даних або покращення загальної узагальненості моделі.

Принцип БЗН полягає в тому, що завдяки паралельному навчання кількох задач одночасно, модель може отримати більш потужну здатність до узагальнення, що є критичним для багатьох реальних застосувань, таких як розпізнавання зображень, обробка природних мов або медична діагностика. Спільне навчання дозволяє також уникати перенавчання на конкретних завданнях, оскільки модель змушена навчатися не лише на конкретному наборі даних, але й у контексті інших завдань, що дає більш широке і стійке розуміння патернів у даних.

З технічної точки зору, багатозадачне навчання може бути досягнуте за допомогою різних архітектур, таких як спільні мережі з декількома вихідними шарами, де кожен з них відповідає за певне завдання, або через використання спільних проміжних представлень для всіх задач.

### 2.2.1 Багатозадачне навчання у нейронних мережах

Попередні дослідження в галузі багатозадачного навчання (БЗН, MTL) визначають два основні типи підходів, що відрізняються способом організації та обробки спільних і специфічних ознак для різних завдань. Перший підхід, відомий як жорсткий обмін параметрами (Hard Parameter Sharing, HPS), передбачає використання спільних шарів для всіх завдань мережі, при цьому вихідні шари є окремими для кожного завдання. Така

архітектура дозволяє значно знизити кількість параметрів, що в свою чергу підвищує ефективність навчання. Другий підхід, м'який обмін параметрами (Soft Parameter Sharing, SPS), припускає наявність окремих мереж для кожного завдання, однак між мережами встановлюється певна узгодженість ваг.

Методи багатозадачного навчання в нейронних мережах були вперше запропоновані Річардом Каруаною у 1998 році, де був описаний підхід HPS [21]. З того часу цей метод набув найбільшого поширення порівняно з SPS. Водночас, на сучасному етапі з'явилися новіші гібридні підходи між HPS і SPS, які поки що не здобули значної популярності.

Попри всі переваги багатозадачного навчання, більшість задач машинного навчання досі вирішуються за допомогою однозадачного навчання (Single Task Learning, STL). Це часто пояснюється складністю реалізації та налаштування багатозадачних архітектур. Навіть у випадках, коли завдання можна поділити на кілька взаємопов'язаних підзадач, традиційно використовуються окремі STL моделі, які навчаються на одному наборі даних незалежно одна від одної. Хоча цей підхід є простішим, він не дозволяє мережам взаємодіяти і обмінюватися інформацією, що обмежує їх здатність використовувати спільні ознаки. Методи БЗН, зокрема HPS та SPS, усувають цей недолік, забезпечуючи ефективніше використання спільних характеристик.

Основний етап реалізації багатозадачного навчання полягає в правильному визначенні набору завдань. Складність цього процесу значною мірою залежить від природи даних і бажаних результатів. У деяких випадках завдання можна без проблем розділити на підзадачі, якщо початкове завдання вимагає класифікації даних у кількох вимірах. Однак у інших випадках це неможливо, і в таких ситуаціях визначаються допоміжні завдання, продуктивність яких не є критично важливою, але вони допомагають покращити результат основного завдання. Як зазначено в дослідженнях, набір завдань має бути взаємопов'язаним для досягнення

оптимальних результатів. Більшість реалізацій потребують одного спільного набору даних для всіх підзадач, хоча новіші підходи можуть припускати використання різних вхідних даних для кожного завдання.

При реалізації HPS зазвичай використовуються спільні шари для виділення загальних ознак та окремі вихідні шари для кожного завдання, як представлено на рисунку 2.1. Зазвичай при створенні багатозадачної мережі застосовуються попередньо навчанні STL моделі, які адаптуються або об'єднуються для створення єдиної мережі. Однак перетворення кількох STL моделей у одну багатозадачну вимагає значних змін в архітектурі, включаючи створення нових шарів для кожного конкретного завдання. Це також потребує налаштування гіперпараметрів, що збільшує складність системи. Крім того, спільна архітектура часто призводить до збільшення часу навчання та потреби у додаткових обчислювальних ресурсах.

З іншого боку, підхід SPS дозволяє використовувати окремі мережі для кожного завдання, при цьому для забезпечення схожості між вагами моделей вводяться функції регуляризації. Це дозволяє зменшити різницю між вагами різних мереж, щоб вони навчалися на схожих ознаках, проте сама мережа залишається більш гнучкою порівняно з HPS, що особливо важливо при роботі з підзадачами, які мають різну складність.

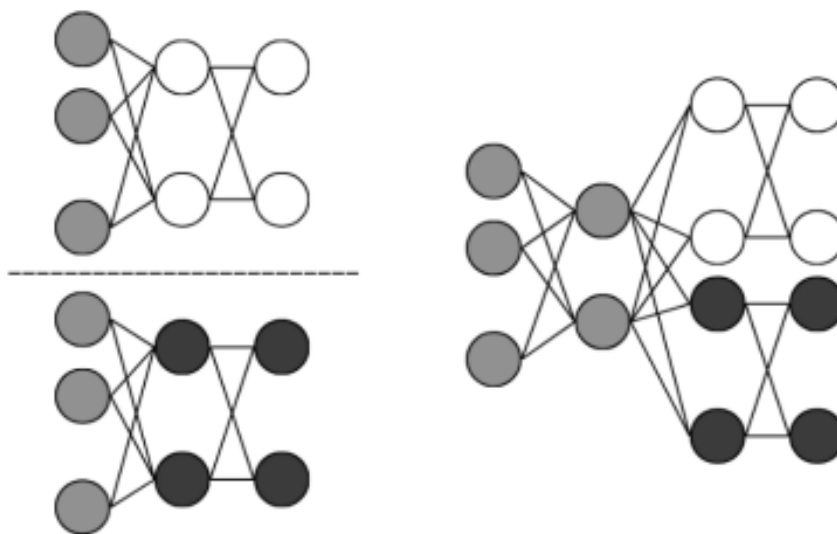


Рисунок 2.1 – Різниця реалізації методу STL з методом HPS

## 2.2.2 Переваги багатозадачного навчання

У своїх роботах Р. Каруана запропонував низку гіпотез, що пояснюють переваги багатозадачного навчання (MTL) порівняно з іншими підходами [19]. Одними з основних переваг MTL є:

- зменшення упередженості. Оскільки багатозадачні мережі одночасно вирішують кілька завдань, вони можуть виявляти загальні ознаки, що є корисними для різних завдань. Це дозволяє мережі не підлаштовуватися виключно під специфічні вимоги однієї задачі, що, в свою чергу, покращує здатність мережі до генералізації на нові дані;

- зменшення впливу шуму. Всі завдання в MTL можуть містити певний рівень шуму, що варіюється для кожної задачі. Однак завдяки спільному навчанню для кількох завдань мережа здатна «усереднювати» цей шум, знижуючи ймовірність перенавчання і підвищуючи стійкість моделі до випадкових варіацій у даних;

- механізм підслуховування. Цей процес полягає в тому, що підмережі можуть взаємодіяти між собою, обмінюючись корисною інформацією, навіть якщо одна з підмереж не здатна самостійно виділити потрібні ознаки. Таким чином, позичаючи інформацію від сусідніх підмереж, система може легше моделювати зв'язки, які не є специфічними для окремої задачі, що сприяє поліпшенню результатів;

- запобігання перенавчанню. Оскільки одні й ті самі ознаки можуть навчатися різними підмережами, це зменшує ймовірність перенавчання, особливо коли одна підмережа надмірно підлаштовується під дані. При перенавченні однієї з підмереж інші підмережі коригують параметри навчання, що допомагає уникнути перенавчання всієї мережі.

Ці властивості підкреслюють переваги багатозадачного навчання як ефективного методу для покращення результатів навчання та зменшення негативних ефектів, властивих традиційним підходам у машинному навчанні.

### 2.2.3 Попередні дослідження в сфері багатозадачного навчання

Концепція багатозадачного навчання (MTL) була вперше детально описана Р. Каруаною в його роботах, зокрема в його дисертації, де була запропонована ідея одночасного навчання кількох задач за допомогою спільних параметрів [9]. У той час, через обмеженість технологій і розмірів моделей, зазвичай використовувались архітектури з мінімальною кількістю прихованих шарів, здебільшого з одним або двома. Проте, з розвитком більш складних моделей, зокрема глибоких нейронних мереж, спостерігається зростаючий інтерес до використання глибоких багатозарових архітектур для виконання задач у багатозадачному навчанні.

Один із прикладів таких інноваційних підходів представлений у дослідженні, яке демонструє ефективність застосування багатозадачного навчання при розпізнаванні людської активності, зокрема для виявлення орієнтирів обличчя, використовуючи згорткові нейронні мережі (CNN). Виявлено, що інтеграція додаткових підзадач, таких як оцінка пози голови або визначення атрибутів обличчя, не лише підвищує точність моделі, але й значно зменшує її обчислювальну складність. Це досягається завдяки спільному навчанні всіх підзадач, що дозволяє зменшити кількість необхідних параметрів і, відповідно, знижує витрати на обчислення.

Крім того, у сучасних дослідженнях розглядаються нові методи, які покращують роботу багатозадачних моделей [9]. Одним з таких підходів є робота, в якій представлено нову методику, що поєднує методи жорсткого та м'якого обміну параметрами в рамках багатозадачного навчання. Цей підхід включає реалізацію структури, схожої на SPS (soft parameter sharing), але з деякими змінами. Наприклад, для шарів з однаковою глибиною запропоновано видалити традиційний регуляризатор. Замість цього використовуються спеціальні блоки, які зшивають виходи різних шарів попіксельно, виконуючи матричні операції для об'єднання даних з різних задач.

Ці нововведення дозволяють забезпечити більш гнучке і ефективне управління інформацією між різними підзадачами. Ключовим елементом такого підходу є параметри, які визначають, скільки інформації передається між різними підмережами. Залежно від значень цих параметрів, інформація може передаватися більш активно або залишатися локальною для кожної конкретної задачі. Це дозволяє досягти кращої адаптації моделі до специфіки кожного завдання, при цьому зберігаючи ефективність в навчанні загальних ознак, які є корисними для всіх задач.

Таким чином, розроблені методи обміну параметрами між підмережами в багатозадачних системах сприяють поліпшенню продуктивності моделей, одночасно зменшуючи потребу в додаткових обчисленнях.

## 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

### 3.1 Основні типи архітектур та їх особливості

#### 3.1.1 Однозадачне навчання

Дослідження багатозадачного навчання та визначення доцільності його застосування передбачає обов'язкову побудову та тестування моделі типу STL (Single-Task Learning). Наявність такої моделі є необхідною для об'єктивного порівняння з підходами MTL (Multi-Task Learning) в однакових експериментальних умовах. Залучення виключно результатів попередніх досліджень є недоцільним через відмінності у застосованих методах, обробці даних і параметрах навчання, що унеможлиблює проведення коректного аналізу.

Тому в межах роботи здійснюється створення і навчання власної STL-мережі, що буде оцінена за єдиною методологією разом із MTL-моделями. Значну увагу також приділено питанню складності та обчислювальної вартості трансформації моделей від STL до MTL. Оскільки підходи багатозадачного навчання зазвичай потребують більших обчислювальних ресурсів, вибір архітектур відбувався з урахуванням балансу між рівнем ефективності та обмеженням обчислювальної складності. У рамках дослідження перевага надавалася тим рішенням, які забезпечували прийнятні результати при збереженні відносної простоти архітектури, навіть якщо складніші варіанти демонстрували вищу якість [10].

Критерії оцінювання включали підсумкові показники ефективності моделей, часові витрати на навчання, а також аналіз динаміки покращення результатів у процесі тренування. Вибір архітектурних рішень був безпосередньо обумовлений особливостями відповідних наборів даних, тому їхній детальний опис буде подано окремо разом із характеристиками використаних датасетів.

### 3.1.2 Жорсткий обмін параметрами

Одним із методів формування мереж багатозадачного навчання є інтеграція кількох базових моделей однозадачного навчання. У випадках, коли підзадачі мають схожу природу, таке об'єднання не становить значної складності. Архітектура HPS (Hard Parameter Sharing) передбачає спільне використання нижніх шарів мережі, тоді як основне завдання полягає у визначенні шарів, які мають бути загальними для всіх підзадач. Після цього для кожної окремої задачі створюються власні верхні шари. Приклад такої архітектури представлено на рисунку 3.1, де білим кольором позначено шари спільного використання, а сірим – індивідуальні компоненти для кожної задачі.

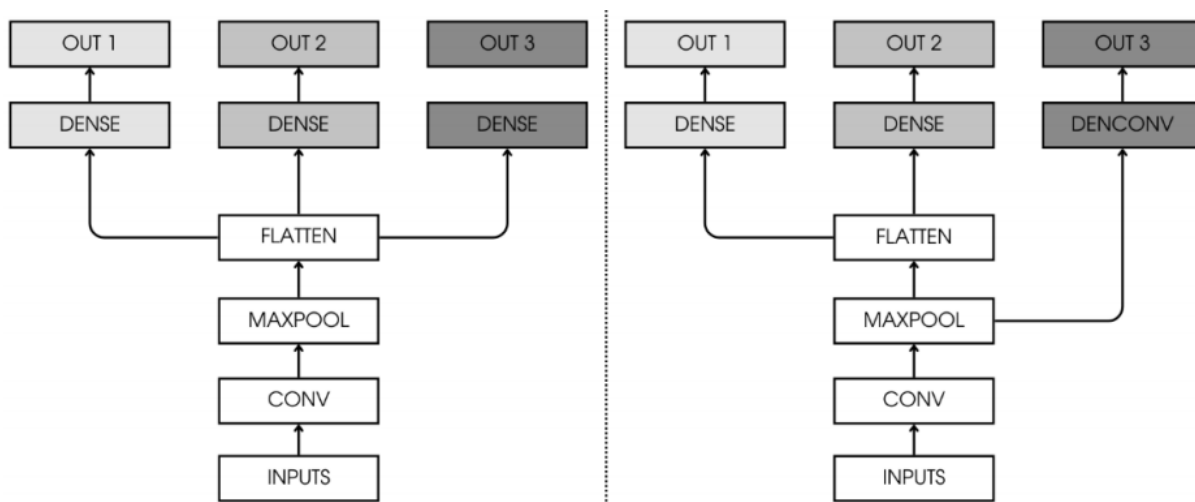


Рисунок 3.1 – Приклад архітектури мережі жорсткого обміну параметрами

Незалежно від характеру задач, вихідні шари кожної підзадачі спрямовані на оптимізацію загальної, а не суто локальної функції втрат. Як правило, сукупна функція втрат визначається як зважена сума індивідуальних функцій втрат для кожної задачі. Нехай  $N$  – кількість задач,  $L_i$  – функція втрат для  $i$ -ої задачі, а  $\alpha_i$  – відповідний ваговий коефіцієнт; тоді загальну функцію втрат можна записати у вигляді [12]:

$$L_{\text{total}} = \sum_{i=1}^N \alpha_i L_i. \quad (3.1)$$

Така побудова дозволяє гнучко враховувати вплив кожної задачі на процес оптимізації, налаштовуючи вагові коефіцієнти вручну або адаптивно в межах навчального процесу.

Слід зазначити, що функції втрат можуть суттєво різнитися між окремими підзадачами. Наприклад, у разі розгляду умовної задачі класифікації, що включає три підзадачі, перші дві можуть бути звичайними задачами багатокласової класифікації, для яких застосовується функція втрат на основі крос-ентропії. Третя підзадача може реалізовувати функціонал автоенкодера, орієнтованого на зменшення розмірності вхідних даних для виявлення ключових ознак та усунення аномалій і шуму.

Однак у межах даного дослідження акцент було зроблено саме на задачах класифікації, що зумовило використання функції категоріальної крос-ентропії для всіх підзадач. Як показано в роботі, ця функція втрат є стандартом у багатокласовій класифікації, де вихідні дані кодуються за допомогою методу one-hot encoding [13]. У цьому випадку вихід моделі представляється у вигляді вектора ймовірностей належності зразка до кожного з класів, причому остаточне рішення приймається на основі найбільшої ймовірності.

Формула обчислення функції втрат крос-ентропії має вигляд [14]:

$$CE(Y, \hat{Y}) = -\sum_{i=1}^N y_i \cdot \log \hat{y}_i. \quad (3.2)$$

### 3.1.3 М'який обмін параметрами

Методологія реалізації м'якого обміну параметрами базується на підході, запропонованому у попередньому дослідженні, проте в цій роботі застосовано його адаптовану версію із збереженням основних ідей [15].

Відмінною рисою методу є створення окремих обчислювальних гілок для кожного завдання, що принципово відрізняє його від традиційної стратегії спільного використання параметрів. Після побудови індивідуальних шарів здійснюється нормалізація їх параметрів із наступною мінімізацією відмінностей між ними. Завдяки регуляризації параметри підмереж набувають певної уніфікованості, хоча й не стають повністю ідентичними. Такий підхід сприяє стандартизації поведінки підмереж без втрати специфіки окремих задач.

Регуляризація у цьому методі здійснюється за допомогою тензорної слідової норми (ТСН), що є узагальненням стандартної слідової норми для багатовимірних тензорних об'єктів. Аналогічно до випадку матриць, де слідова норма визначається як сума їхніх сингулярних значень, ТСН розглядає суму слідових норм для усіх можливих режимів тензора [18].

З метою обчислення ТСН попередньо виконується перетворення тензорної структури у двовимірний формат. Серед наявних методів такої трансформації можна виокремити розкладання за Такером, тензорно-потоківі розкладання, а також метод LFA [19]. У даній роботі обрано підхід Такера, який забезпечує розкладання тензора на окремі матриці шляхом фіксації одного з вимірів, після чого обчислюється відповідна слідова норма для кожної з отриманих матриць.

Регуляризаційний член у цьому випадку може бути математично поданий таким чином:

$$\|W\| = \sum_{i=1}^m \alpha_i \|W^{(i)}\|, \quad (3.3)$$

де  $m$  – кількість вимірів тензора;

$\alpha_i$  – ваговий коефіцієнт, що регулює внесок кожного режиму у загальну форму;

$\|W^{(i)}\|$  – сідлова норма розгорнутої матриці відповідного режиму.

Параметри  $\alpha_i$  можуть варіюватися залежно від специфіки задачі та структури тензора, однак у даному дослідженні для всіх режимів використано фіксоване значення  $\alpha_i = 0.01$ , що дозволяє досягти збалансованості між основною функцією втрат і регуляризаційним компонентом.

Загальна функція втрат для задачі з множиною завдань  $T$  включає як стандартний критерій оптимізації, так і регуляризаційний додатак. Вона визначається наступною формулою:

$$L_{\text{total}} = a_{\text{TNN}} \|W\| + \sum_{t \in T} L_{\text{CE}}, \quad (3.4)$$

де  $a_{\text{TNN}}$  – коефіцієнт ваги регуляризаційного члена;

$\|W\|$  – значення тензорної слідової норми;

$\sum_{t \in T} L_{\text{CE}}$  – сумарна функція втрат категоріальної крос-ентропії для всіх

задач.

### 3.1.4 Мережі для пересічного навчання

Метод пересічного зшивання (Cross-Stitch) є інтегративним підходом до багатозадачного навчання, що вперше був запропонований у дослідженні [20]. Основна концепція полягає у досягненні балансу між автономністю обробки кожного завдання та можливістю взаємного обміну релевантною інформацією між відповідними підмережами. Згідно з цією ідеєю, кожне завдання має власну окрему обчислювальну гілку, проте за допомогою механізму пересічного зшивання передбачається контрольований обмін ознаками, що потенційно покращує загальну якість навчання.

У практичній реалізації даного підходу між шарами підмереж вставляються спеціальні блоки зшивання, які виконують функцію адаптивних регуляторів інформаційних потоків. Такі блоки здійснюють зважене об'єднання вихідних ознак різних завдань шляхом навчання вагових коефіцієнтів, які визначають ступінь передачі інформації від однієї підмережі до іншої. Кожна ознака окремо піддається оптимізації за допомогою механізму зваженого множення, що дозволяє формувати більш релевантне представлення перед передачею даних до наступного шару.

Графічна інтерпретація принципу роботи окремого елемента пересічного зшивання подана на рисунку 3.2, що демонструє механізм комбінування ознак за допомогою навчуваних вагових множників. Цей механізм дозволяє мережі гнучко регулювати інтенсивність обміну інформацією між завданнями на рівні окремих шарів, забезпечуючи адаптивне налаштування залежно від конкретних вимог до задач.

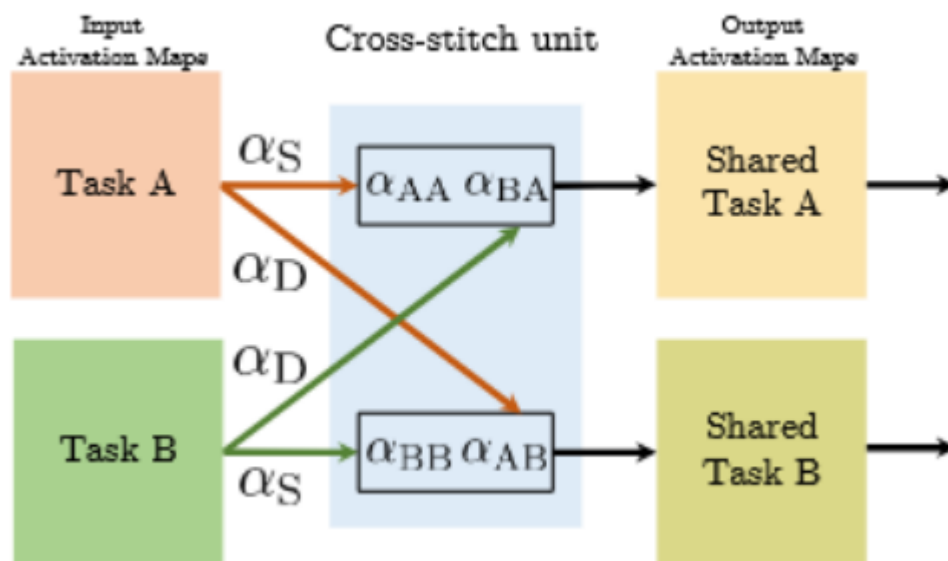


Рисунок 3.2 – Представлення окремого елемента з використанням пересічного зшивання

Математичне підґрунтя роботи механізму пересічного зшивання було детально розглянуто у попередньому розділі, зокрема у формулі (3.4), де

описано спосіб формування зважених сум ознак різних підмереж. Завдяки такій побудові модель отримує можливість самостійно визначати необхідний ступінь взаємодії між завданнями, що істотно знижує ризик перенесення нерелевантної інформації.

Таким чином, метод пересічного зшивання забезпечує значно вищий рівень гнучкості в організації спільного навчання порівняно з традиційними підходами до багатозадачного навчання, де спільне використання параметрів або повна ізоляція підмереж зазвичай не дозволяють досягти оптимального балансу між об'єднанням і розділенням завдань [21].

## 3.2 Використані набори даних для дослідження

### 3.2.1 Набору даних Opportunity

Датасет Opportunity був вперше представлений у 2010 році та спрямований на розв'язання задачі розпізнавання людської активності (HAR), де дані були зібрані за допомогою носимих сенсорів та датчиків, що фіксують параметри навколишнього середовища [22]. Хоча існують й інші загальнодоступні набори даних для подібних задач, вони часто не забезпечують достатньої різноманітності для детального дослідження опортуністичної активності. Серед таких можна згадати PlaceLab, що зосереджується на вивченні навколишнього середовища та об'єктів, набір даних Van Kasteren, що містить записи на довші періоди часу, і Дармштадтський рутинний датасет, спрямований на виявлення патернів неконтрольованої активності, зібраної за допомогою системи Porgurine [23]. Унікальність Opportunity полягає в його специфічному спрямуванні на розвиток опортуністичного підходу до розпізнавання активності людини, що передбачає здатність адаптуватися до різноманітних умов і джерел інформації.

Структурно цей набір даних складається з серії експериментальних прогонів, де учасники виконують різні завдання в середовищі, обладнаному великою кількістю сенсорів. Опис структури і змісту набору можна знайти в оригінальному дослідженні [18]. Експериментальне середовище імітує квартиру, яка включає ліжко, міні-кухню та стіл, де для забезпечення принципів опортуністичної класифікації була створена мережа датчиків, що дозволяє ефективно поєднувати різні датчики залежно від сценарію. Така архітектура сенсорної мережі дає можливість моделювати різноманітні ситуації під час подальшої класифікації. Деталі налаштування лабораторії та розміщення датчиків ілюструються на рисунку 3.3.

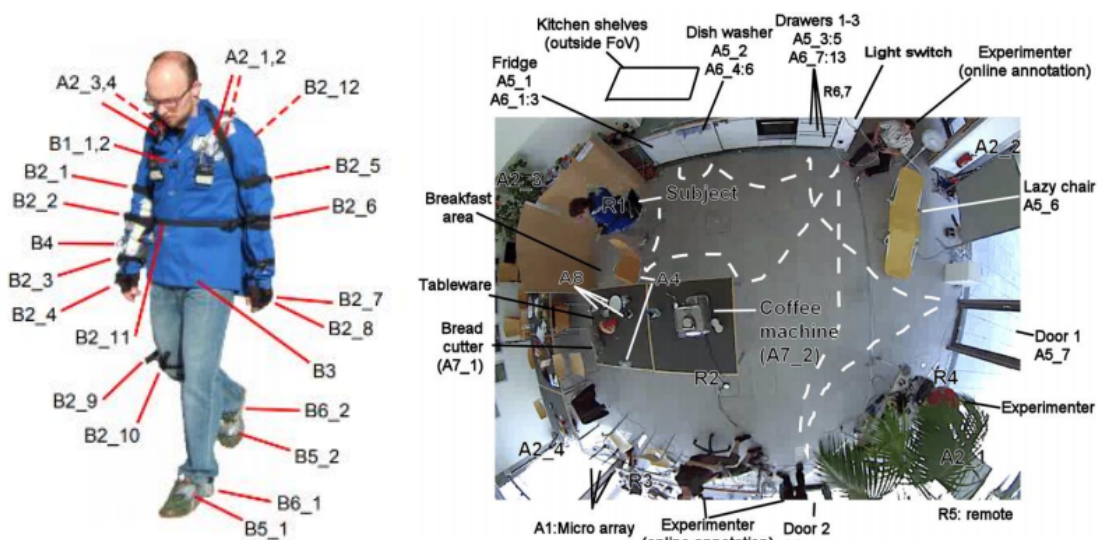


Рисунок 3.3 – Розміщення сенсорів на тілі людини для дослідження

Датчики у наборі Opportunity можна розподілити на дві основні категорії:

– тілесні датчики, що включають прискорювачі, інерційні вимірювальні прилади (що фіксують положення), системи відносного позиціонування (визначають відстань між руками та тілом), а також мікрофони на тілі учасця. Усього таких каналів 133;

– навколишні датчики, що включають акселерометри, геркони (для визначення відкриття/закриття дверей та шухляд), камери та системи для надширокосмугового позиціонування. Загалом є 109 таких датчиків.

Дані з різних сенсорів були синхронізовані і оброблені з частотою 30 Гц, після чого вони були вручну марковані за допомогою п'яти або семи категорій активності. Ці категорії включають Locomotion (типи руху: стояння, ходьба, сидіння, лежання), HL\_Activity (активності високого рівня, такі як «відпочинок», «прибирання», «час на каву»), LL\_Left\_Arm та LL\_Right\_Arm (рухи відповідних рук), LL\_Left\_Arm\_Object та LL\_Right\_Arm\_Object (взаємодія рук з об'єктами), а також ML\_Both\_Arms (комбінація активності обох рук). Автори Opportunity рекомендують використовувати категорію ML\_Both\_Arms для класифікації, і це підтверджується кращими результатами в попередніх дослідженнях [24].

Що стосується специфікації даних, кожен канал може містити значення NULL, що не завжди вказує на втрату даних, але може бути результатом специфіки поведінки учасників або технічних проблем, таких як оклюзія сигналу чи розряджені батареї. Оцінки авторів вказують на те, що близько 2.5% даних було втрачено через збої в з'єднанні, а також переповнення спектра внаслідок великої кількості сенсорів, які працюють в діапазонах 2.4 ГГц і 5 ГГц.

Датасет Opportunity містить чотири сеанси, кожен з яких складається з шести прогонів. Перші п'ять прогонів включають повсякденні дії, а шостий прогін – спеціальне завдання Drill, де учасник виконує коротшу послідовність дій. Один прогін ADL займає від 15 до 20 хвилин, а прогін Drill – від 20 до 35 хвилин. Загальний обсяг даних сягає близько 155 хвилин при частоті дискретизації 30 Гц.

Один з основних етапів роботи з набором даних включає підготовку даних для навчання нейронної мережі, що передбачає попередню обробку даних, включаючи очищення від пропущених значень, шумів та аномалій, а

також перетворення та масштабування даних. Важливим етапом при роботі з часовими рядами є сегментація даних, що дозволяє полегшити подальший аналіз, зменшуючи кількість параметрів для навчання мережі та оптимізуючи час навчання. Для датасету Opportunity сегментація здійснюється шляхом розбиття даних на двосекундні відрізки, кожен з яких містить 64 зразки. Під час цього процесу мітки активності призначаються згідно з принципом більшості для кожного сегмента.

Як показано на рисунку 3.4, обробка даних передбачала також виявлення і корекцію аномалій, що були виявлені в деяких каналах. Ці аномалії зазвичай мали невеликий обсяг, але їх корекція була важливою для покращення якості даних для подальшої класифікації.

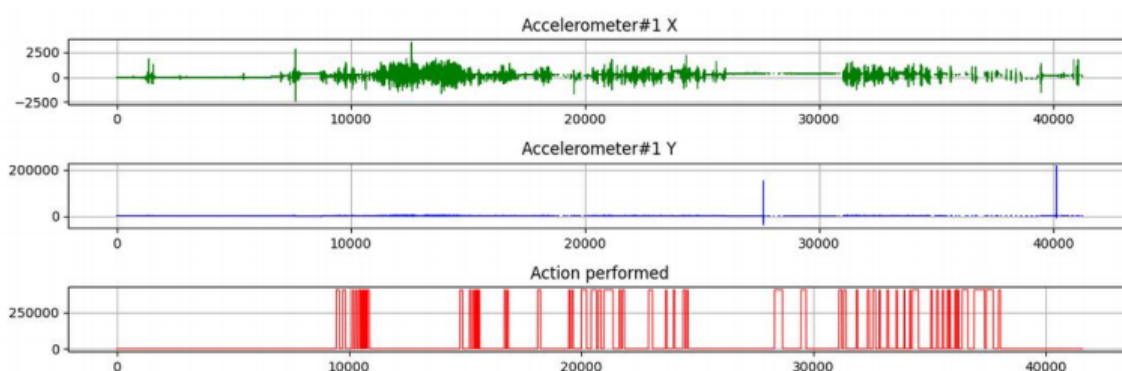


Рисунок 3.4 – Аналіз аномалій для набору даних Opportunity

### 3.2.2 Опис набору даних Dear

Основними наборами даних, що використовуються для аналізу емоцій на основі фізіологічних сигналів, є Dear (Database for Emotion Analysis using Physiological Signals) та SEED (SJTU Emotion EEG Dataset). Обидва орієнтуються на дані, зібрані із сенсорних пристроїв, насамперед електроенцефалографії (ЕЕГ). Однак суттєва різниця між ними полягає в тому, що SEED ґрунтується виключно на ЕЕГ, тоді як Dear доповнюється даними електроокулографії (ЕОГ), електроміографії (ЕМГ), вимірюваннями

електропровідності шкіри, дихання та температури тіла випробуваного. Загалом у Dear задіяно 40 сенсорних каналів, серед яких 32 належать до ЕЕГ. Завдяки цьому датасет Dear є більш придатним для досліджень у сфері багатозадачного навчання [25].

У дослідженні брали участь 32 особи (16 чоловіків і 16 жінок) віком від 20 до 38 років, які оцінювали емоції, викликані відеокліпами. Одним із головних викликів при формуванні датасету було визначення системи категоризації емоцій.

Наразі в науковій спільноті відсутній консенсус щодо класифікації емоцій, і обговорення цього питання залишаються актуальними. Виділяють три основні підходи до класифікації емоцій.

Перший базується на визначенні шести базових емоцій: радості, смутку, гніву, огиди, страху та здивування, з яких походять інші комбінації емоцій. Деякі варіації цієї концепції пропонують альтернативні набори основних емоцій, тоді як ієрархічні моделі класифікації не набули широкого поширення [5].

Другий підхід використовує шкальні методи оцінювання емоцій, зокрема емоційне коло Плачтика або модель валентності та збудження Рассела [27]. Саме останній підхід було обрано для Dear. Відповідно до нього, емоції розташовуються у двовимірному просторі, тобто валентність відображає спектр від негативних до позитивних емоцій, а збудження – рівень активації від спокою до збудження. За потреби додається третя шкала – домінування, що охоплює діапазон від почуття безпорадності до повного контролю.

Третій підхід представлений моделлю компонентних процесів, яка хоча й рідше застосовується, усе ще враховується у відповідних дослідженнях [28].

У підсумку було обрано модель основного афекту для оцінки емоцій, де учасників просили оцінити валентність, збудження та домінування за шкалою від 1 до 9. Використання шкал із манекенами SAM, представлених

на рисунку 3.5, дозволяло респондентам обирати будь-яке значення на континуумі. Окрім цього, реєструвалася знайомість із відео (оцінка від 1 до 5) та ступінь сподобання кліпу (від 1 до 9).

Одним із основних мотивів використання складної багатозадачної моделі класифікації замість простої регресійної моделі стала варіативність оцінок між респондентами, обумовлена індивідуальними відмінностями у сприйнятті та знайомістю з відеоматеріалами.

Це призводить до розбіжностей в оцінках емоційних станів навіть при перегляді однакових відео.

Класифікація емоцій у двовимірному просторі за моделлю Рассела зазвичай призводить до їх розподілу на чотири квадранти з високою/низькою валентністю та збудженням [29]. Це зумовлює вирішення задачі через дві окремі проблеми багатокласової класифікації.

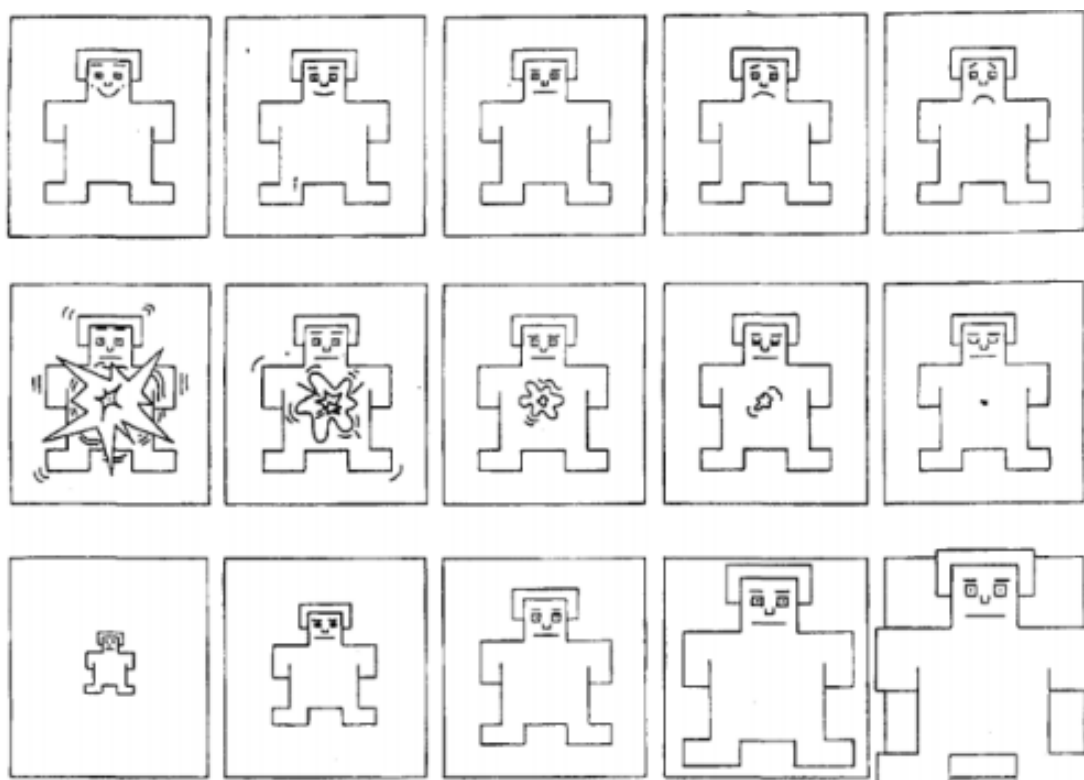


Рисунок 3.5 – Манекени для самооцінки, зверху вниз: валентність, збудження, домінування

Попередні підходи переважно базувалися на однозадачному навчанні із роздільними моделями для кожного виміру емоцій. Однак через очікувану кореляцію між збудженням та валентністю застосування багатозадачного навчання (MTL) може підвищити ефективність моделі.

На відміну від датасету Opportunity, дані Dear вже доступні в обробленому вигляді. Проте для даного дослідження виникла необхідність у додаткових етапах обробки, зокрема у присвоєнні міток до кожного зразка через відсутність часової зміни міток, на відміну від Opportunity. Було прийнято рішення присвоювати однакову мітку всім кадрам одного запису, після чого трансформувати оцінки у бінарні класи за допомогою розподілу навколо медіанного значення [30].

Сегментація також вимагала додаткової уваги через суб'єктивний характер оцінок. Ставлення учасників до відео чи жанру впливало на результати оцінювання, що ускладнювало завдання класифікації.

Більше того, як вказано в літературі, неінвазивна ЕЕГ обмежена у точності при фіксації нейронної активності, що відповідає за емоції, оскільки більшість таких процесів відбувається у глибоких структурах мозку, недоступних для поверхневого зчитування [8]. Це робить критичним правильний вибір і розподіл даних на навчальні та тестові підмножини.

У рамках дослідження розглянуто два основних варіанти розподілу даних. Підхід, залежний від суб'єкта, де навчальні дані включають записи всіх учасників, а тестові дані формуються шляхом випадкового вибору прикладів. Цей підхід дозволяє моделі виявляти шаблони для кожного суб'єкта, але обмежує її здатність узагальнювати на нових осіб.

Підхід, незалежний від суб'єкта, передбачає, що модель навчається на даних певної групи учасників, тоді як тестування відбувається на записах інших осіб, яких не було в тренувальній вибірці. Такий метод забезпечує об'єктивнішу оцінку здатності моделі до узагальнення, оскільки вона повинна правильно інтерпретувати патерни емоційних відповідей нових респондентів. Незважаючи на більшу складність, саме цей варіант розподілу

даних краще відповідає реальним умовам використання моделей емоційної класифікації у практичних застосуваннях, де передбачити особливості нових користувачів заздалегідь неможливо [16].

У даному дослідженні обрано підхід незалежний від суб'єкта, що дозволило моделю краще оцінити узагальнюючі можливості побудованої системи. При цьому особлива увага приділялася методам сегментації даних і правильному балансуванню класів, щоб мінімізувати потенційні перекося у навчальній вибірці.

Крім того, для посилення стійкості моделі до індивідуальних відмінностей випробуванців використовувалися різні техніки підготовки даних, включаючи нормалізацію показників за кожним учасником окремо та стандартизацію ознак для забезпечення однакової шкали їх значень.

Таким чином, у роботі було враховано ключові виклики, пов'язані зі специфікою емоційних даних, зокрема суб'єктивність сприйняття, обмеження електроенцефалографії як інструменту збору сигналів та необхідність розробки стратегій розподілу даних для забезпечення реалістичної оцінки продуктивності моделей у нових умовах.

### 3.3 Вибір архітектури для дослідження

#### 3.3.1 Опис архітектури для набору даних Opportunity

Наступним етапом після попередньої обробки даних стало визначення підходящих методів багатозадачного навчання (MTL) для роботи з підготовленими наборами даних. На цьому етапі було створено базові нейронні мережі окремо для кожного датасета, які в подальшому могли бути модифіковані з використанням різних технік MTL. Базові архітектури обиралися на основі аналізу попередніх досліджень, однак перевага надавалася простішим рішенням через обмеження у часі та ресурсах програмного забезпечення. З огляду на певні теоретичні припущення [16],

передбачалося, що застосування MTL навіть до простих моделей має забезпечити результати, принаймні не гірші, ніж при використанні складніших архітектур у режимі одиничного завдання (STL).

Як було зазначено раніше, важливою основою у сфері виявлення людської активності слугує дослідження [16], де було здійснено систематичне порівняння різних підходів класифікації з використанням датасету Opportunity. Ця робота слугувала орієнтиром як при підготовці даних, так і при пошуку оптимальної базової моделі. Основні методи глибинного навчання, що розглядалися в [16], були детально описані у підрозділі 2.1. Інші методи, зокрема автоенкодері, техніки побудови ознак за допомогою Code-Book та ручне виокремлення ознак, не розглядалися у межах цього дослідження через їхню непридатність до парадигми багатозадачного навчання.

Оцінювання якості моделей здійснювалося за трьома метриками: точністю, середньою F1-оцінкою та зваженою F1-оцінкою. Для валідації виділялася приблизно чверть усіх доступних даних. Точність визначалася як частка правильних передбачень серед усіх прогнозів. Формально її можна записати у вигляді:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \delta(\hat{y}_i, y_i), \quad (3.5)$$

де  $\delta$  – функція, що дорівнює 1 у разі правильного передбачення та 0 в іншому випадку;

$N$  – загальна кількість прикладів;

$\hat{y}_i, y_i$  – передбачені та істинні мітки відповідно.

Крім того, для більш збалансованого оцінювання використовувався показник F1, що є гармонійним середнім точності та пригадування:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.6)$$

У задачах багатокласової класифікації обчислення F1 здійснювалося за підходами макроусереднення, мікроусереднення та зваженого усереднення. Зокрема, макроусереднення дозволяє однаково враховувати кожен клас незалежно від кількості його прикладів, мікроусереднення базується на загальному об'єднанні правильних передбачень для всіх класів, а зважене усереднення компенсує дисбаланс класів, беручи до уваги їхню відносну кількість у вибірці.

У межах побудови базових моделей були розглянуті три основні архітектури: згорткова нейронна мережа (CNN), довга короткочасна пам'ять (LSTM) та їх комбінація – гібридна CNN-LSTM. За результатами дослідження [17] було встановлено, що гібридні архітектури демонструють найкращі показники. Водночас у межах експериментів було зафіксовано, що прості CNN-моделі при належній регуляризації, зокрема із додаванням шарів Dropout, здатні забезпечувати продуктивність, що не поступається гібридним рішенням, а в окремих випадках навіть перевищує їх (рисунок 3.6).

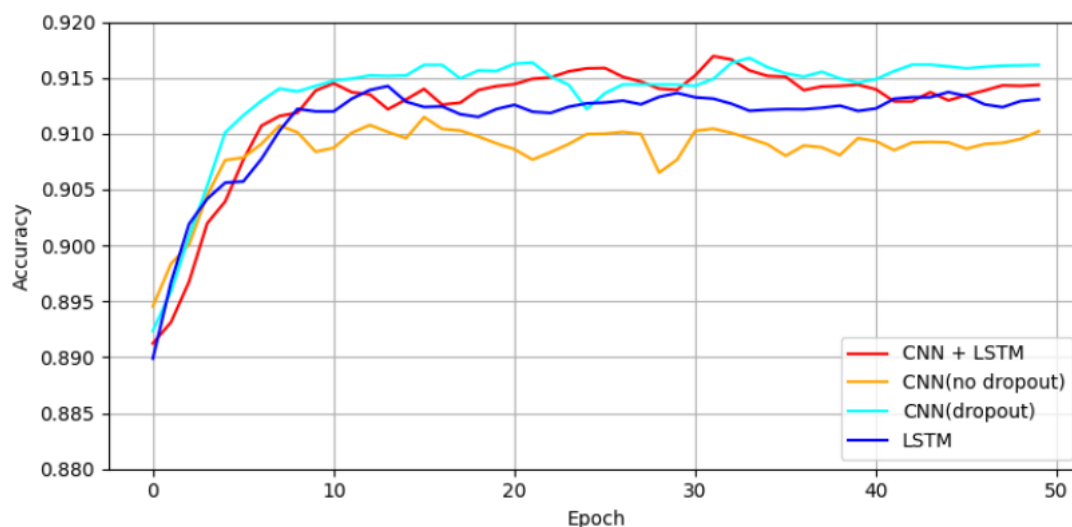


Рисунок 3.6 – Графік точності при тренуванні базових моделей

### 3.3.2 Опис архітектури для набору даних Dear

Архітектура мережі для роботи з датасетом Dear була схожа на вибір для датасету Opportunity. Більшість попередніх підходів зводиться до застосування методів екстракції ознак перед навчанням, при цьому часто використовуються тільки дані з ЕЕГ, які піддаються поділу на різні частотні діапазони за допомогою перетворення Фур'є, методів на основі вейвлетів або просто статичних ознак [17].

Для цієї роботи було обрано використання згорткової нейронної мережі (CNN), без рекурентних елементів. У процесі експериментів було виявлено, що ефективність мережі значною мірою залежить від того, яка частина даних була використана для навчання, зокрема при завантаженні даних через процедуру їх перемішування. Оскільки було вирішено уникати завантаження всіх елементів у пам'ять одночасно на етапі навчання, було використано буфер певного розміру, який попередньо завантажував елементи, перш ніж вони потрапляли до мережі. Після цього на вхід подавалися випадкові елементи з цього буферу, з подальшою заміною кожного елемента на наступний з датасету. Такий підхід дозволив значно зменшити негативний ефект перемішування даних під час навчання, при цьому зберігаючи позитивний ефект цього процесу.

На відміну від датасету Opportunity, де застосовувалась функція категоріальної крос-ентропії, у даному випадку для навчання використовувалися функція втрат типу розрідженої категоріальної крос-ентропії та розріджена категоріальна точність.

### 3.4 Характеристика та особливості технічної реалізації

Для реалізації обраних методів багатозадачного навчання було використано мову програмування Python, що забезпечує значні переваги завдяки простоті синтаксису та наявності потужних бібліотек, як для

машинного навчання, так і для загальних обчислень. Python підтримує паралельні обчислення за допомогою бібліотек CUDA, CuDNN і NumPy, що дозволяє використовувати графічні процесори (GPU) для значного прискорення тренування великих нейронних мереж.

Основними інструментами для створення та тренування нейронних мереж є бібліотеки Keras і TensorFlow. TensorFlow є низькорівневою бібліотекою, що надає інструменти для роботи з тензорами, автоматичного диференціювання та оптимізації, хоча її використання вимагає більшої уваги до деталей програмування. Keras є високоабстрактним API, яке дозволяє швидко будувати моделі з мінімальною кількістю коду, при цьому використовуючи TensorFlow як основну платформу для виконання операцій.

Для обробки даних в TensorFlow зазвичай використовують NumPy, оскільки бібліотека дозволяє зберігати та обробляти дані у вигляді багатовимірних масивів. Однак, коли дані передаються до моделі через функцію `model.fit`, вони спочатку передаються у вигляді повного масиву, що може створити значне навантаження на оперативну пам'ять, особливо при роботі з великими датасетами, як у випадку з Opportunity і Dear. Це може призвести до надмірної витрати пам'яті та навіть до збоїв в роботі системи.

Для вирішення цієї проблеми була використана нова версія TensorFlow 2.0, що підтримує формат `tf.dataset`. Це дозволяє значно зменшити вимоги до пам'яті, використовуючи модуль `tf.data` для ефективного завантаження та обробки даних, а також формат серіалізації даних Protocol Buffer (`protobuf`) для компактного зберігання та передачі даних. Використання форматів TFRecord та `protobuf` дозволяє ефективно зберігати та обробляти великі набори даних.

Для забезпечення ефективного завантаження даних в оперативну пам'ять, після серіалізації даних необхідно провести процес десеріалізації, що дозволяє динамічно змінювати форму тензорів та знижує потребу в зберіганні всіх елементів датасету в пам'яті одночасно. Процес

перемішування та розбиття на батчі здійснюється після етапу десеріалізації за допомогою асинхронного буферизування, що дозволяє зберігати поточний батч та кілька наступних у пам'яті.

При роботі з датасетом Deep розмір буферу був визначальним параметром, що впливає на продуктивність мережі. Початковий розмір буферу, рівний 1000 елементів, не давав задовільних результатів, тому для покращення точності було збільшено розмір буферу. Для подальшого вдосконалення результатів датасет був попередньо переміщений на етапі передобробки за допомогою методів бібліотеки NumPy.

Мережі для різних методів, таких як HPS та SPS, були реалізовані окремо, а потім об'єднані під час навчання. У випадку з CSN, мережа була побудована повністю, що було необхідно через особливості її архітектури. Важливою проблемою при розробці мережі стала необхідність вирішення проблеми розриву графу мережі, для чого вихідні дані паралельних підмереж об'єднувалися перед передачею через блок CrossStitch, а потім негайно розподілялися на початкові тензори для подальшого використання.

### 3.5 Представлення розроблених архітектур

Після створення нейронні мережі були візуалізовані за допомогою веб-застосунку Netron, який забезпечує детальне зображення архітектури мереж у вигляді графічних моделей із повним описом кожного шару, їхніх назв та розмірностей. Зокрема, на рисунку 3.7 зображено архітектуру мережі STL для датасетів Opportunity та Deep.

Як було зазначено раніше, основною архітектурою для реалізації моделей було обрано згорткову нейронну мережу з розширеною структурою. На рисунку видно дві різні архітектури, які мають мінімальні відмінності між собою. Обидві архітектури включають блок нормалізації перед першим згортковим шаром, що значно покращує процес навчання, зменшуючи ймовірність виникнення проблеми зникаючих градієнтів. Після

кожного згорткового шару застосовується операція максимального об'єднання, що дозволяє зменшити розміри простору даних, зберігаючи найбільш важливу інформацію. Така послідовність операцій повторюється кілька разів, що дає змогу мережі формувати багаторівневе представлення вхідних даних.

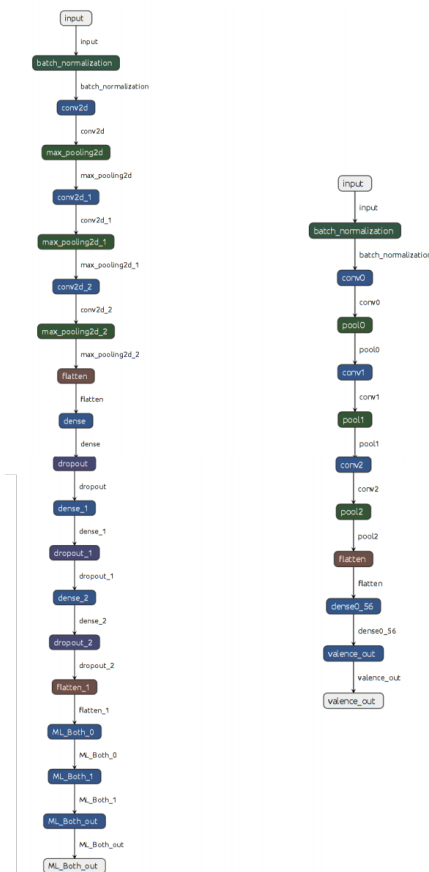


Рисунок 3.7 – Представлення архітектури SLN для розглянутих датасетів

Основною відмінністю між двома архітектурами є використання в мережі Opportunity додаткових повнозв'язних шарів (Dense), в порівнянні з одним таким шаром у мережі Deep, а також наявність шарів Flatten, що перетворюють багатовимірні дані в одномірні вектори. Крім того, мережа Opportunity містить шари Dropout, яких немає в мережі Deep. Це сприяє боротьбі з перенавчанням, що не було виявлено при навчанні на датасеті

Dear. Таким чином, відмінності між архітектурами зумовлені, в основному, кількісними та якісними характеристиками даних.

Архітектури мереж HPS для обох датасетів мають схожість з архітектурами STL. Підхід HPS реалізовано на основі базових згорткових нейронних мереж, де замість спільних Dense шарів після шару Flatten реалізовано підмережі з індивідуальними повнозв'язними шарами та окремими виходами для кожної підзадачі. Така архітектура з подвійним виходом дозволяє мережам прогнозувати два різні параметри, використовуючи спільні низькорівневі ознаки, виділені згортковими шарами, але з окремими трактуваннями на вищих рівнях обробки. Архітектури мереж HPS для Opportunity та Dear наведено на рисунку 3.8.

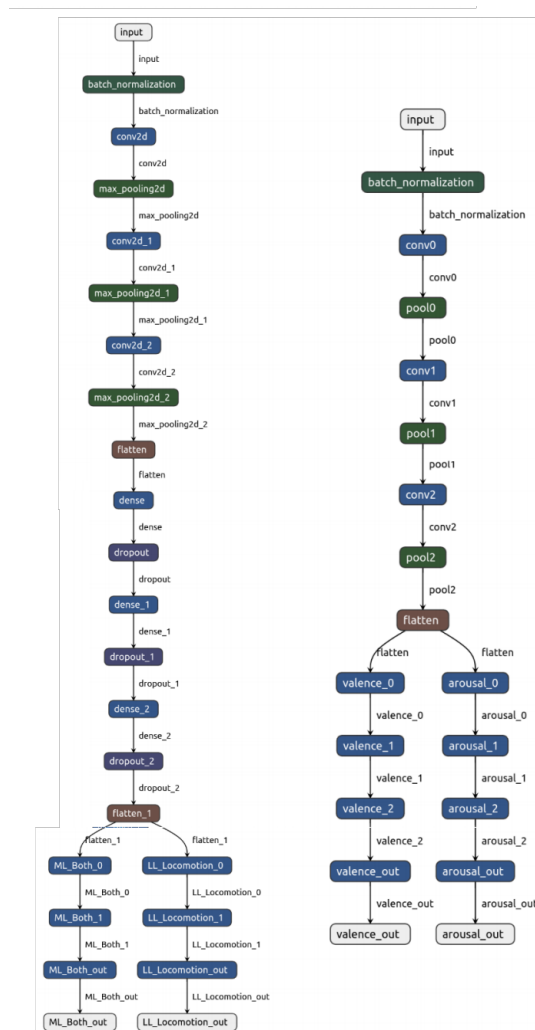


Рисунок 3.8 – Представлення архітектури HPS для розглянутих датасетів

На рисунку 3.9 продемонстровано архітектуру мережі SPS, яка реалізує процес паралельного навчання між двома підмережами.

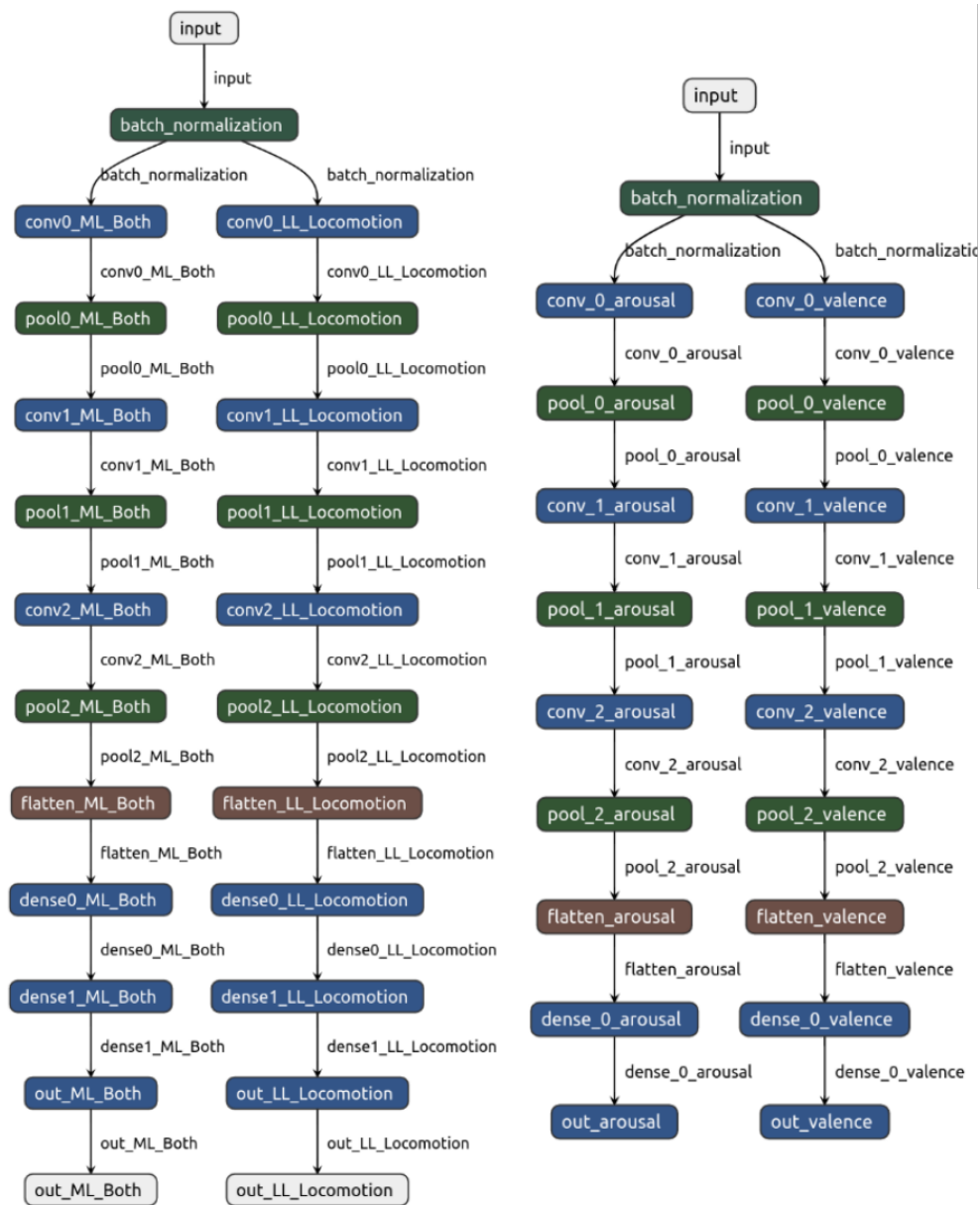


Рисунок 3.9 – Представлення архітектури SPS для розглянутих датасетів

Хоча підмережі створено на основі базової згорткової нейронної мережі, між мережами HPS та SPS існують певні відмінності. Основною відмінністю є відсутність шарів Dropout та спільних Dense шарів у мережі Opportunity, які присутні після першого шару Flatten в мережі HPS. Підмережі Dear у мережі SPS зберігають ідентичність з базовою

архітектурою. Мережа SPS починається з спільного шару нормалізації, після чого розгалужується на дві підмережі, кожна з яких має свої шари Dense для розв'язання окремої підзадачі (рисунок 3.10).

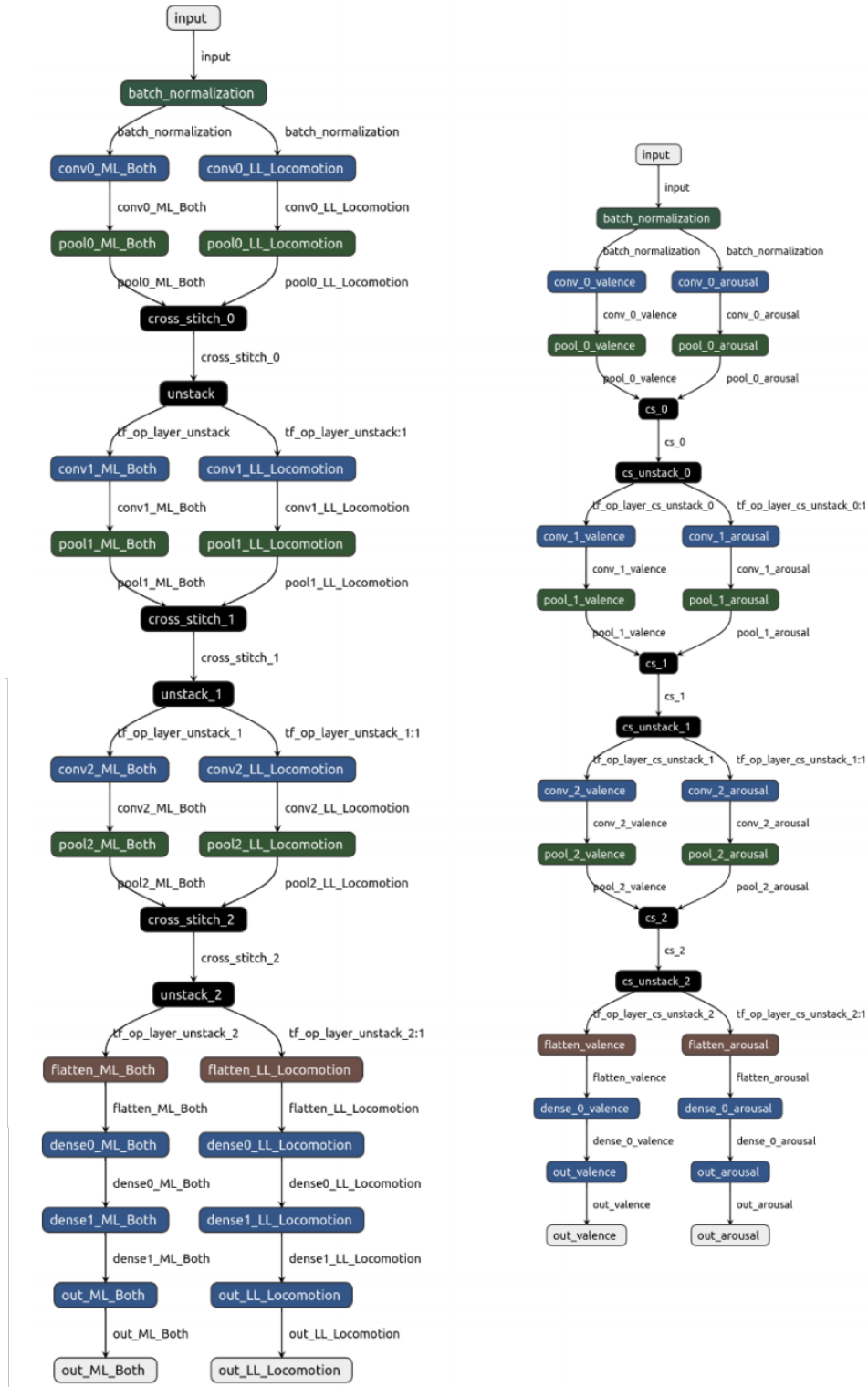


Рисунок 3.10 – Представлення архітектури SPS для розглянутих датасетів

Мережа CSN реалізована подібно до мережі SPS, але з додаванням блоків зшивання, що сприяють обміну інформацією між підмережами. Архітектура починається з шару нормалізації, після чого розгалужується на дві підмережі. Після кожної пари згортки та об'єднання шарів застосовуються шари Cross-stitch і Unstack. Така архітектура формує цикл обміну інформацією: спочатку ознаки обробляються окремо, потім комбінуються через блок зшивання і після цього знову розділяються для подальшої обробки. Така структура повторюється кілька разів, що дозволяє мережі на різних рівнях абстракції вирішувати, які ознаки краще обробляти спільно, а які окремо.

## 4 РЕЗУЛЬТАТИ ПРОВЕДЕНОГО ДОСЛІДЖЕННЯ

### 4.1 Опис процесу оцінки моделі

Як зазначалося раніше, загальноприйнятий підхід до оцінювання та порівняння результатів, що полягає у співставленні отриманих даних із результатами, наведеними у науковій літературі або попередніх дослідженнях, є недостатньо надійним. Такий підхід часто не забезпечує повноцінного відображення особливостей роботи моделей, що призводить до потенційних похибок в інтерпретації їх ефективності. Незважаючи на можливу подібність результатів, отриманих за допомогою моделей STL і MTL, вагомим значення набувають додаткові фактори, зокрема обсяг часу, необхідного для навчання, кількість епох до досягнення цільових показників якості, а також характер динаміки оновлення вагових коефіцієнтів у процесі навчання. Оцінити ці аспекти найбільш адекватно можливо шляхом графічного аналізу перебігу навчання, що і становить основну мету цього розділу.

У процесі навчання після завершення кожної епохи здійснювався розрахунок ключових метрик, зокрема категоріальної точності та показника F1. Для обчислення цих метрик використовувалися функціональні можливості бібліотек Keras та sklearn. Механізм зворотних викликів, реалізований у Keras, дозволяв автоматизувати процес реєстрації та збереження результатів обчислень у відповідних журналах.

Враховуючи стохастичну природу процесу оптимізації, емпіричні залежності показників навчання від кількості епох демонстрували загальну тенденцію, але супроводжувалися значними коливаннями навколо трендових ліній. Для підвищення наочності представлених результатів було застосовано метод локального поліноміального регресійного згладжування (LOESS), що дозволяє апроксимувати тренд шляхом

побудови локальних регресійних моделей. Такий підхід забезпечує стійкість до наявності аномальних значень та випадкових флуктуацій у даних.

Для кожної моделі було побудовано графіки, що відображають динаміку зміни метрик у межах останніх 10 епох для датасету Opportunity та останніх 25 епох для датасету Deep. На основі отриманих результатів також було сформовано порівняльні таблиці, які систематизують характеристики досліджуваних моделей.

## 4.2 Жорсткий обмін параметрами

### 4.2.1 Набір даних Opportunity

У межах дослідження було обрано датасет Opportunity, який охоплює сім різних задач класифікації. Центральною задачею виступило розпізнавання даних з каналу ML\_Both\_Arms, вибір якого ґрунтується на попередніх результатах аналізу, де було виявлено його високу ефективність у вирішенні поставленої проблеми. Таким чином, головна увага була зосереджена на оптимізації процесу класифікації саме для цього каналу.

У першому експерименті додатковим завданням було залучено канал LL\_Locomotion, що відображає базові характеристики рухової активності людини. Ця задача передбачає класифікацію за п'ятьма основними класами та додатковою категорією NULL, що дає змогу врахувати стани, не пов'язані із визначеними діями.

Порівняння проводилося між стандартною архітектурою однозадачного навчання (STL) та модифікованою багатозадачною архітектурою (MTL), реалізованою у вигляді згорткової нейронної мережі із двома підмережами, що розгалужуються після шару типу dense. Таке розділення дозволяло моделі вивчати ознаки, характерні для кожного завдання окремо. У процесі перетворення базової архітектури на багатозадачну було особливою увагою приділено налаштуванню ваг функцій

втрат для кожного завдання та кількості шарів, спільних для всіх задач, відповідно до підходу HPS.

Другий експеримент передбачав подальшу модифікацію архітектури: спільні шари типу dense були вилучені, а до кожної підмережі після першого шару додано шар регуляризації dropout із ймовірністю випадіння 0.4. Метою змін було оцінити вплив архітектурних особливостей на ефективність багатозадачного навчання та визначити, які рішення сприяють покращенню загальної продуктивності мережі.

Результати першого експерименту наведено на рисунку 4.1. Для оцінки якості класифікації використовувалися дві метрики, а саме макро-F1 та мікро-F1. Мікро-F1 агрегує показники класифікації на основі суми істинно позитивних, хибно позитивних та хибно негативних результатів без урахування розподілу класів, тоді як макро-F1 забезпечує рівнозначний вклад кожного класу, що робить його більш придатним для аналізу незбалансованих даних. Як видно з рисунку 4.1, архітектура багатозадачного навчання демонструє перевагу над однозадачною за мікро-F1 показником, тоді як за макро-F1 результати залишаються на порівнянному рівні. Видалення спільних шарів негативно впливало на якість класифікації, що свідчить про необхідність використання загальних репрезентацій у підмоделях. Додатково було виявлено, що за умов однакової важливості обох завдань модель досягала кращих кінцевих результатів, тоді як акцент на основній задачі спричиняв поступове зниження якості в процесі навчання.

У другому експерименті аналізувалося, як кількість допоміжних задач впливає на ефективність багатозадачного навчання. Було розглянуто три варіанти: навчання з двома задачами, з чотирма задачами, а також із залученням усіх наявних каналів. Як показано на рисунку 4.2, додавання трьох допоміжних задач сприяло підвищенню якості класифікації порівняно з двома завданнями.

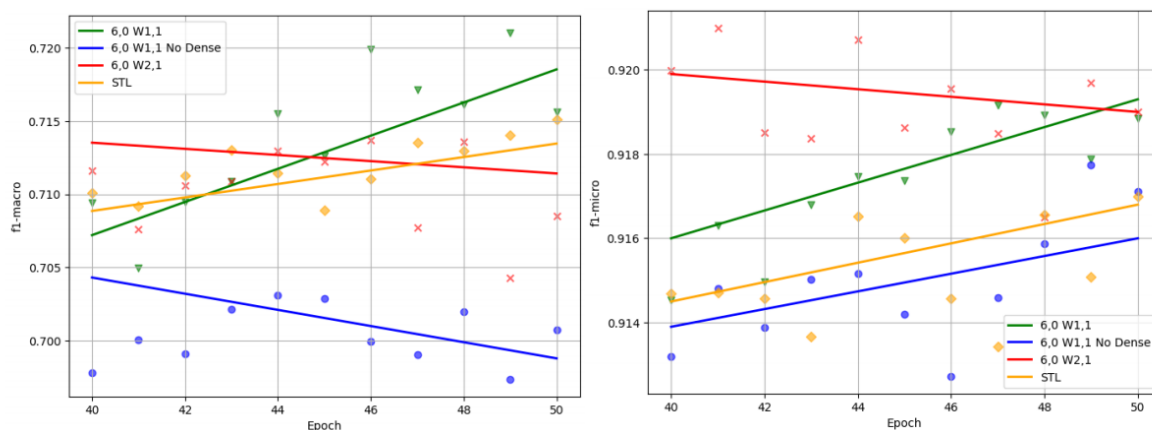


Рисунок 4.1 – Графіки порівняння точності архітектур STL і HPS

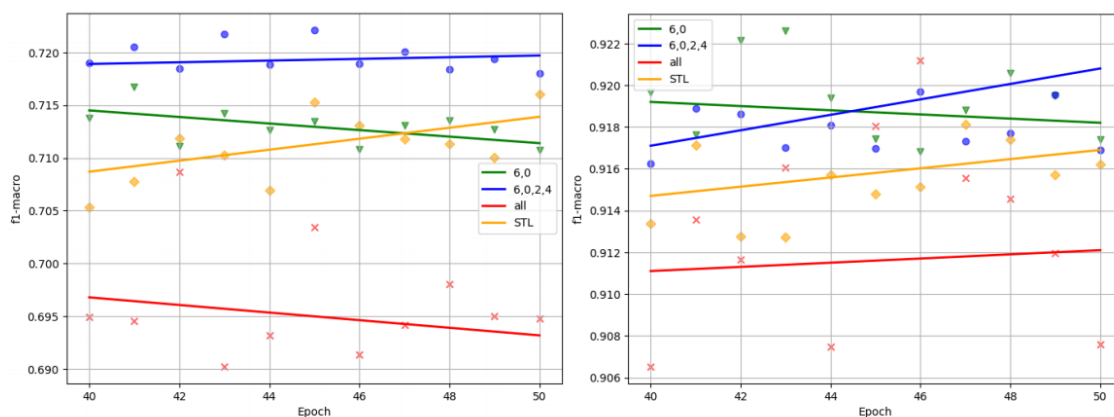


Рисунок 4.2 – Графіки порівняння кількості завдань для архітектури HPS

Водночас залучення всіх каналів зумовило зниження результативності, що підтверджує важливість ретельного відбору допоміжних задач при побудові ефективних архітектур багатозадачного навчання.

#### 4.2.2 Набір даних Deep

У межах дослідження також було виконано експериментальні дослідження на основі датасету Deep. Процедура проведення експериментів залишалася аналогічною до попередніх, однак специфіка полягала у виборі

задач: класифікації рівня збудження та валентності. Обидві задачі розглядалися як рівнозначні за важливістю, що зумовило використання єдиної метрики точності для оцінювання ефективності моделей.

На рисунку 4.3 наведено результати порівняння архітектур: однозадачного навчання (STL), багатозадачного навчання із загальними шарами (HPS) та варіанту із повністю розділеними підмережами. Аналіз отриманих даних свідчить, що архітектура із використанням спільних шарів демонструє найвищу якість класифікації. Водночас стандартна однозадачна мережа показала кращі результати, ніж модель з повністю відокремленими шарами.

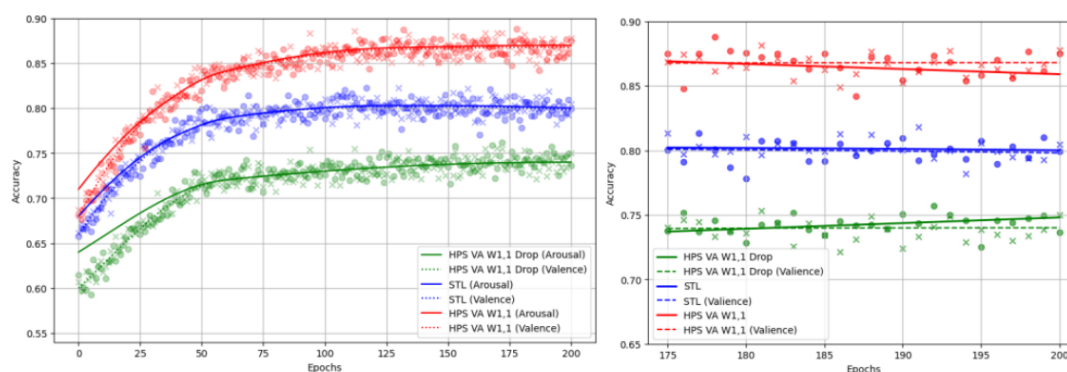


Рисунок 4.3 – Графіки порівняння архітектур STL та HPS

Така тенденція узгоджується з висновками, отриманими під час експериментів на датасеті Opportunity, підтверджуючи доцільність застосування спільного представлення ознак у багатозадачних мережах для підвищення їх загальної ефективності.

#### 4.2.3 Порівняльний аналіз жорсткого обміну параметрів для розглянутих наборів даних

У процесі дослідження на основі набору даних Opportunity було зафіксовано незначне покращення показників ефективності, що

підтверджується результатами статистичного аналізу із використанням 95% довірчого інтервалу, наведеними на рисунку 4.3 та у таблиці 4.1. Відсутність перетину довірчих інтервалів між результатами методів STL та HPS свідчить про статистично значуще покращення продуктивності при застосуванні багатозадачного підходу. Отримані дані також вказують на потенціал подальшого підвищення якості класифікації за рахунок оптимізації гіперпараметрів моделі.

У випадку застосування методів до набору даних Dear спостерігалось покращення результатів майже на 6% порівняно з однозадачним навчанням, що є суттєвим досягненням і свідчить про перспективність використання багатозадачного підходу в цьому контексті.

Особливої уваги заслуговує той факт, що архітектура HPS продемонструвала найкращі результати серед усіх розглянутих варіантів, що є нетиповим висновком для даного класу задач. Простота архітектурної організації HPS, яка передбачає спільне використання параметрів між підзадачами без можливості динамічного налаштування, водночас забезпечує певні переваги та породжує обмеження. Зокрема, статичний характер спільного представлення обумовлює критичну залежність якості навчання від правильного вибору підзадач для об'єднання.

Складність підбору оптимальних гіперпараметрів при побудові багатозадачної мережі на основі базових однозадачних моделей залишається вагомим викликом. Незважаючи на зафіксоване покращення результатів, повний потенціал архітектури HPS на сьогодні залишається реалізованим не повністю. Значна чутливість методу до початкових умов моделювання вказує на необхідність подальших досліджень, спрямованих на розробку більш ефективних процедур налаштування гіперпараметрів.

Актуальні наукові розробки спрямовані на зниження залежності моделей від ручного налаштування через впровадження методів оцінювання невизначеності для автоматизованого коригування вагових коефіцієнтів, а також використання алгоритмів із адаптивним керуванням швидкістю

навчання для кожної підзадачі окремо. Зазначені підходи створюють передумови для суттєвого підвищення ефективності архітектур типу HPS та їх подальшого впровадження у практичні системи обробки даних.

### 4.3 М'який обмін параметрами

#### 4.3.1 Набір даних Opportunity

На відміну від методу жорсткого обміну параметрів, експериментальні дослідження із використанням м'якого обміну проводилися лише для двозадачної архітектури. Це було зумовлено істотним зростанням кількості параметрів, які необхідно оптимізувати, що призводило до значного збільшення обчислювальних витрат при спробі навчання моделі одночасно на чотирьох задачах. Враховуючи обмеження ресурсів та продуктивності, проведення таких експериментів було визнано недоцільним у межах даного дослідження.

Однак, як видно з рисунку 4.4, усі варіації моделі із застосуванням м'якого обміну (SPH) продемонстрували результати, що суттєво поступалися стандартному підходу однозадачного навчання (STL).

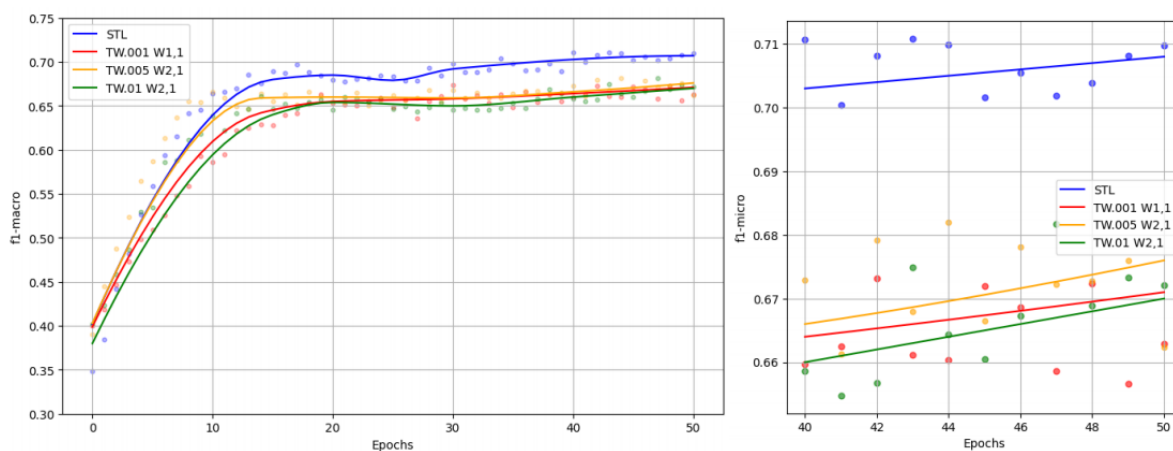


Рисунок 4.4 – Графіки порівняння архітектур STL та SPS

Ключовою відмінністю підходу м'якого обміну є введення додаткового гіперпараметра – норми тензорного сліду, а також відповідної ваги його впливу на загальну функцію втрат. Початково для цього параметра було використано рекомендоване в попередніх дослідженнях значення 0.001, яке у ході експериментів підлягало коригуванню. За результатами емпіричного налаштування оптимальним було визнано значення норми, що дорівнює 0.005.

#### 4.3.2 Набір даних Dear

Експериментальні дослідження на датасеті Dear були організовані за аналогічною методологією, що і для набору даних Opportunity. Проте результати вкотре підтвердили менш ефективну роботу методу м'якого обміну у порівнянні з традиційним однозадачним навчанням. Незважаючи на те, що за графічними результатами при значенні норми тензорного сліду 0.001 модель SPH демонструвала певні покращення, процес тренування був достроково зупинений через надмірні витрати часу та ресурсів. Зокрема, навчання протягом 100 епох тривало близько 25 годин, що було оцінено як малоефективне з огляду на співвідношення витрат і досягнутих результатів (рисунок 4.5)

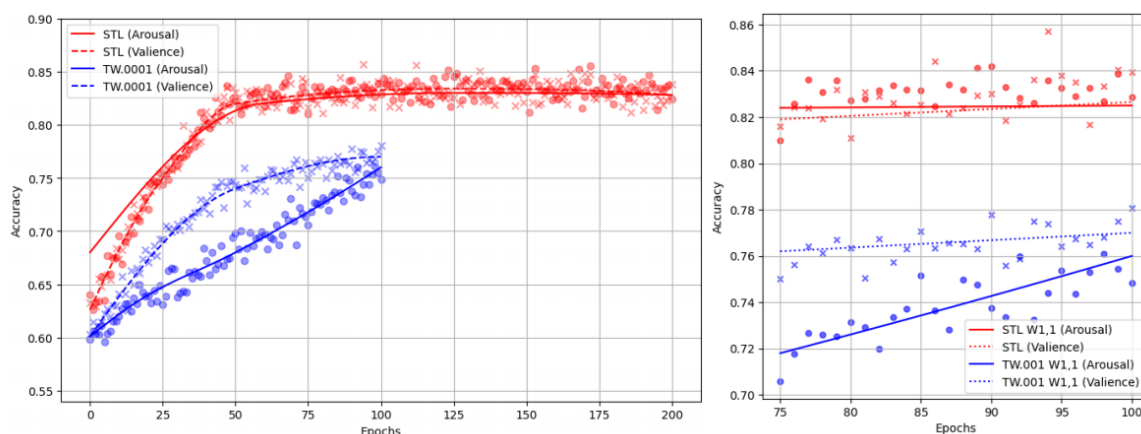


Рисунок 4.5 – Графіки порівняння архітектур STL та SPS

### 4.3.3 Порівняльний аналіз м'якого обміну параметрів для розглянутих наборів даних

Архітектурна концепція SPS ґрунтується на принципі паралельного навчання кількох моделей зі схожою структурою для різних завдань, без прямого обміну інформацією між ними. Взаємна схожість параметрів підмереж досягається за рахунок застосування регуляризаційних механізмів, що стимулюють їхню конвергенцію.

У межах цього дослідження як регуляризатор використовувалася тензорна слідова норма. Проте її впровадження не забезпечило покращення результатів на обох розглянутих наборах даних, що було дещо неочікувано з огляду на успішні застосування аналогічного підходу в інших прикладних областях. Одним із можливих пояснень цього є специфіка обробки часових рядів, оскільки попередні успішні експерименти здебільшого проводилися на синтетичних або низьковимірних реальних даних.

Необхідність подальшого аналізу підтверджується тим, що методи HPS та SPS концептуально є близькими, однак отримані результати виявилися суттєво відмінними. Крім того, спостережено, що ослаблення впливу або повне виключення тензорної слідової норми з регуляризації призводить до покращення точності на датасеті Deep, де середня точність класифікації для валентності склала  $0.823 \pm 0.004$ , а для збудження –  $0.812 \pm 0.012$ . Водночас, на наборі Opportunity точність знизилася майже на 30%.

Отже, основні труднощі, ймовірно, пов'язані із особливостями використання або реалізації обраної форми регуляризації. Подальші дослідження мають бути спрямовані на оцінювання ефективності альтернативних регуляризаційних підходів, зокрема використання класичних L1- та L2-норм або їхніх комбінацій, що може відкрити нові можливості для підвищення якості навчання моделей.

#### 4.4 Мережа пересічного зшивання

Як і в випадку з дослідженням SPH, експерименти були обмежені двома завданнями. Архітектура мережі включала три послідовні блоки зшивання, кожен з яких мав матричну структуру розміром  $2 \times 2$ . Така конфігурація забезпечувала оптимальний компроміс між складністю моделі та її здатністю ефективно обмінюватися інформацією між підзадачами.

Особливо важливим аспектом дослідження CPS стала можливість візуалізації процесів обміну інформацією за допомогою теплових карт. Цей метод дозволяє моніторити інтенсивність обміну даними між різними шарами мережі та виявляти закономірності і взаємозалежності в процесі навчання. Візуалізація еволюції взаємодії між завданнями на різних рівнях абстракції відкриває нові можливості для аналізу та оптимізації архітектури багатозадачних нейронних мереж.

##### 4.4.1 Набір даних Opportunity

Основним параметром для порівняння ефективності мережі була кількість нейронів у шарах dense. У першому шарі кількість нейронів була стабільною та дорівнювала 1000, тоді як у другому шарі вона змінювалася в межах від 50 до 500. Результати цього експерименту представлено на рисунку 4.6.

З отриманих результатів видно, що багатозадачний підхід не зміг перевищити однозадачний. Це пояснюється тим, що мережа CPS є варіантом мережі SPH, яка не є оптимальною для вирішення подібних задач. Оцінка теплових карт показала, що більша частина інформації не була ефективно передана між підмережами. Незважаючи на те, що в останньому блоці обмін інформацією був значно більший порівняно з попередніми, цього виявилось недостатньо для досягнення найкращих результатів при використанні цього підходу.

Згідно з результатами, представленими на рисунку 4.7, багатозадачний підхід досі не перевищує ефективність однозадачного підходу.

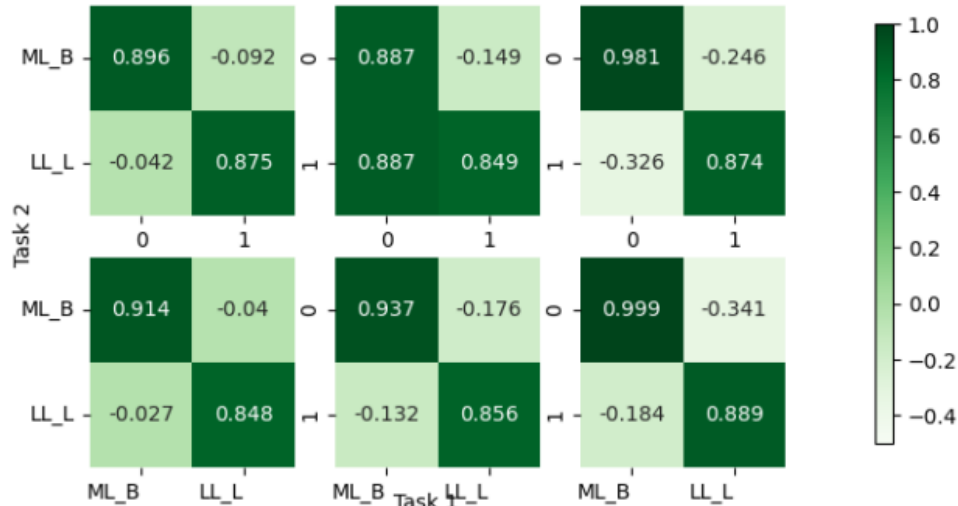


Рисунок 4.6 – Візуалізація процесу обміну параметрами з блоків CS

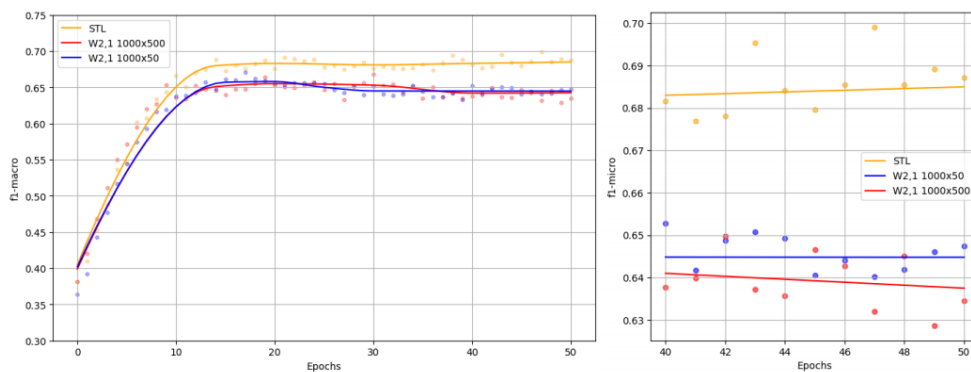


Рисунок 4.7 – Графіки порівняння архітектур STL та CSN

Це можна пояснити тим, що архітектура мережі CPS, яка є варіацією мережі SPN, не є оптимальною для вирішення таких завдань. Крім того, аналіз теплових карт показує, що значна частина інформації не була передана між підмережами. Хоча в останньому блоці обмін інформацією між підмережами зріс порівняно з попередніми етапами, його обсяг все ще

не був достатнім для досягнення максимальної ефективності використання цього підходу.

#### 4.4.2 Набір даних Dear

Однак, під час експериментів з використанням датасету Dear було виявлено, що підхід CPS перевищує ефективність однозадачного навчання, як показано на рисунку 4.8. При цьому, аналізуючи теплові карти, можна відзначити ще одну відмінність у порівнянні з датасетом Opportunity – обмін даними здійснювався протягом усього навчального процесу. Хоча кількість переданої інформації залишалася відносно невеликою, це вказує на потенціал для покращення архітектури з метою подальшого вдосконалення її використання (рисунок 4.9).

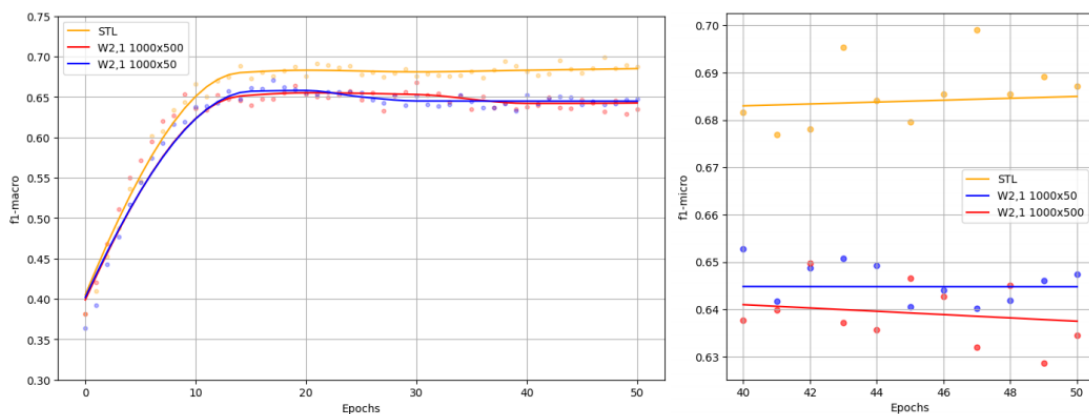
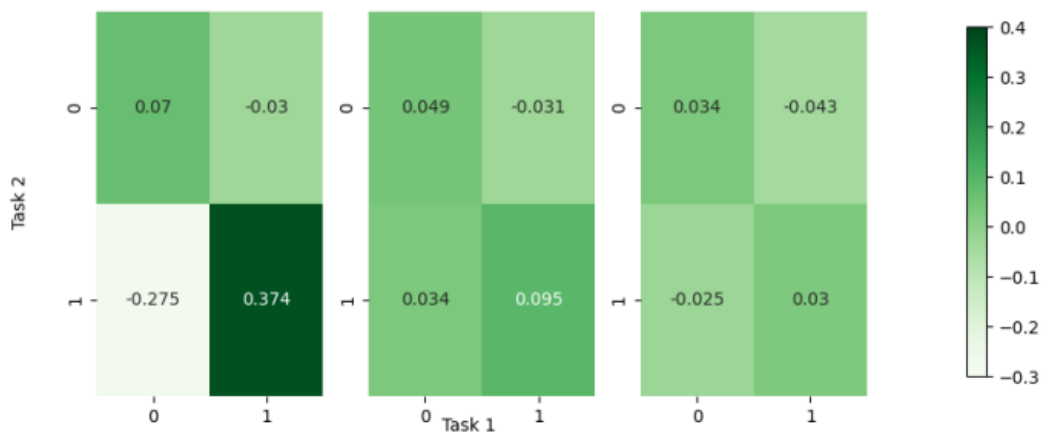


Рисунок 4.8 – Графіки порівняння архітектур STL та CSN



## Рисунок 4.9 – Візуалізація процесу обміну параметрами 3 блоків CS

### 4.4.3 Порівняльний аналіз мережі пересічного зшивання параметрів для розглянутих наборів даних

Під час проведення експериментів на наборі даних Opportunity, багатозадачний підхід CPS не зміг продемонструвати перевагу над однозадачним навчанням. Як показано на рисунку 4.7, за точністю класифікації на цьому наборі даних мережа однозадачного навчання показала стабільніші результати. Це можна пояснити тим, що мережа CPS, яка є варіацією мережі SPH, не є оптимальною для обробки даних цього типу, зокрема для часових рядів, які характеризуються високою залежністю між елементами в часі. Анатомія мережі CPS передбачає взаємодію між кількома підмережами, проте через складність завдань обмін інформацією між підмережами виявився обмеженим. Теплові карти, що демонструють процес передачі інформації між підмережами, показали, що більшість інформації не була передана належним чином, а отже, алгоритм не зміг ефективно використовувати переваги багатозадачності. Хоча обмін інформацією в останньому блоці навчання був значно більшим, його все одно було недостатньо для досягнення оптимальних результатів.

На наборі даних Dear підхід CPS продемонстрував значно кращі результати порівняно з однозадачним навчанням. Як видно на рисунку 4.8, точність класифікації для багатозадачної мережі була вищою, що свідчить про ефективніший обмін інформацією між підмережами в процесі навчання. Це особливо помітно в тому, що під час навчання даних відображалась активна передача інформації між підмережами через теплові карти. В порівнянні з Opportunity, на Dear інформаційний обмін тривав протягом всього процесу навчання, що вказує на кращу здатність моделі взаємодіяти між завданнями. Проте кількість переданої інформації все ще залишалася обмеженою, що вказує на необхідність подальших вдосконалень. У той час

як процес обміну був більш активним, його ефективність не досягала оптимального рівня, що свідчить про потребу в доопрацюванні архітектури CPS для підвищення результатів на цьому наборі даних.

#### 4.5 Порівняльний аналіз розглянутих підходів

Загалом, результати експериментів на датасеті Dear були значно кращими, ніж на датасеті Opportunity, що підтверджується даними в таблиці 4.1. Важливим фактором при оцінці ефективності моделей також є час, необхідний для навчання, особливо коли використовуються обмежені обчислювальні ресурси, як це було в цьому дослідженні. Для датасету Opportunity результати виявилися менш оптимістичними. Різниця між методами STL та MTL виявилася або мінімальною, що робить застосування MTL недоцільним, або достатньо великою на користь STL, що вказує на переваги однозадачного підходу.

Таблиця 4.1 – Порівняння методів для набору даних Opportunity

Підхід	Варіант	F1-macro ± CI	Час навчання (г)
STL	-	0.711 ± 0.004	2.12
HPS	W1,1	0.718 ± 0.003	2.21
	W2,1	0.712 ± 0.003	2.16
	W2,1 No Dense	0.690 ± 0.002	2.26
	W2,1,1,1	0.718 ± 0.003	2.30

	Усі завдання	$0.691 \pm$ 0.003	2.45
--	-----------------	----------------------	------

Продовження таблиці 4.1

SPS	W1,1	$0.675 \pm$ 0.005	04.05
	W2,1	$0.670 \pm$ 0.007	3.98
CSN	W2,1 1000x50	$0.640 \pm$ 0.003	4.60
	W2,1 1000x500	$0.635 \pm$ 0.003	4.45

Однак, на датасеті Dear результати виявилися значно більш обнадійливими, як показано в таблиці 4.2. Тут використання MTL, зокрема через метод HPS, призвело до значного покращення результатів. Проте, метод SPS показав значно гірші результати як за точністю класифікації, так і за часом навчання. В цілому, результати показали, що застосування методів багатозадачного навчання (MTL) є ефективним лише в певних випадках, коли можна коректно визначити підзавдання, які взаємно доповнюють одне одного.

Таблиця 4.2 – Порівняння методів для набору даних Dear

Підхід	Варіант	Точність $V \pm CI$	Точність $A \pm CI$	Час навчання (г)
STL	-	$0.802 \pm$ 0.008	$0.807 \pm$ 0.005	5.3
HPS	W1,1	$0.850 \pm$ 0.003	$0.867 \pm$ 0.003	5.8

	W1,1	0.752 ±	0.740 ±	
	Drop	0.006	0.005	5.7

Продовження таблиці 4.2

SPS	TW0.0001	0.770 ± 0.005	0.760 ± 0.008	26.1 (102 еп.)
	TW0.001	0.410 ± 0.002	0.565 ± 0.003	15.2 (55 еп.)
CSN	W1,1	0.833 ± 0.003	0.849 ± 0.003	6.2

Отже, загальний висновок цього дослідження полягає в тому, що ефективність використання методів багатозадачного навчання сильно залежить від конкретного датасету. У випадку, коли завдання підходять для багатозадачної оптимізації, методи MTL можуть демонструвати кращі результати. Проте для деяких типів задач, таких як Opportunity, однозадачні підходи можуть бути більш ефективними.

## ВИСНОВКИ

У даній роботі досліджено підхід до розпізнавання діяльності людини за допомогою багатозадачного навчання на основі глибинних нейронних мереж, зокрема за допомогою мереж пересічного зшивання та обміну параметрами між підзадачами. Проведений аналіз показав, що традиційні методи розпізнавання діяльності людини, які ґрунтуються на класичних алгоритмах комп'ютерного зору або однотипних нейронних мережах, не завжди можуть ефективно враховувати різноманіття та складність завдань у реальному часі. Багатозадачні моделі на основі глибинних нейронних мереж, зокрема з використанням пересічного зшивання, здатні більш ефективно виявляти загальні ознаки для кількох завдань, що дозволяє підвищити точність і знизити складність моделі.

Дослідження концепції багатозадачного навчання показало, що поєднання підзадач, таких як оцінка пози голови та визначення атрибутів обличчя, дозволяє значно покращити точність та зменшити обчислювальні витрати. Використання механізму обміну параметрами між підмережами забезпечує ефективне перенесення знань між задачами, що також сприяє підвищенню загальної ефективності моделі.

У ході експериментальних досліджень продемонстровано, що застосування запропонованого підходу дозволяє досягти високих результатів у розпізнаванні діяльності людини в реальному часі. Результати показали, що використання багатозадачного навчання з елементами обміну параметрами дозволяє не тільки підвищити точність прогнозування, але й значно зменшити витрати на обчислювальні ресурси.

Запропонований підхід має великий потенціал для практичного застосування в таких сферах, як системи безпеки, моніторинг людської діяльності в розумних містах, а також у медицині для моніторингу пацієнтів та спортивному аналізі. Використання глибинних нейронних мереж з багатозадачним навчанням та обміном параметрами може забезпечити

високу точність та оперативність при розпізнаванні різних видів діяльності людини.

Перспективи подальших досліджень полягають у розробці нових гібридних моделей багатозадачного навчання, що поєднують різні методи оптимізації та розширення підходу для розпізнавання діяльності людини в складних, багатовимірних середовищах. Розширення досліджень у напрямку покращення ефективності адаптації моделей до нових умов та інтеграції додаткових підзадач дозволить ще більше розкрити потенціал багатозадачних нейронних мереж у реальному часі.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P. 1251–1258. DOI: 10.1109/CVPR.2017.195.
2. Devlin J., Chang M. W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). 2019. P. 4171–4186. DOI: 10.18653/v1/N19-1423.
3. Dosovitskiy A., Beyer L., Kolesnikov A., et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale // *International Conference on Learning* URL: <https://openreview.net/forum?id=YicbFdNTTy> (дата звернення: 05.05.2025).
4. Hagan M. T., Menhaj M. Training Feedforward Networks with the Marquardt Algorithm // *IEEE Transactions on Neural Networks*. 1994. Vol. 5, No. 6. P. 989–993. DOI: 10.1109/72.329697.
5. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90.
6. Raffel C., Shazeer N., Roberts A. et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*. 2020. Vol. 21, No. 140. P. 1–67. URL: <http://jmlr.org/papers/v21/20-074.html> (дата звернення: 09.05.2025).
7. Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need. *Advances in Neural Information Processing Systems (NIPS)*. 2017. P. 5998–6008. URL: <https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html> (дата звернення: 15.05.2025).
8. Goodfellow I. J., Pouget-Abadie J., Mirza M. et al. Generative Adversarial Nets. *Advances in Neural Information Processing Systems* 27 (NIPS

2014).2014.P.2672-2680.

URL:

[https://papers.nips.cc/paper\\_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html](https://papers.nips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html) (дата звернення: 01.06.2025).

9. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Vol. 521. P. 436–444. DOI: 10.1038/nature14539.
10. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*. 2015. Vol. 61. P. 85–117. DOI: 10.1016/j.neunet.2014.09.003.
11. Heaton J. Deep learning for time series forecasting: Predict the future with MLPs, CNNs and LSTMs in Python. Heaton Research, 2021. (ASIN для Kindle Edition: B08V8K5APV).
12. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS)*. 2012. P. 1097–1105. URL: <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html> (дата звернення: 18.05.2025).
13. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift // *International Conference on Machine Learning (ICML)*. 2022. P. 448–456. URL: <http://proceedings.mlr.press/v37/ioffe15.html> (дата звернення: 19.05.2025). (Також arXiv:1502.03167).
14. Kingma D. P., Ba J. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. 2015. arXiv:1412.6980. URL: <https://arxiv.org/abs/1412.6980> (дата звернення: 21.05.2025).
15. Srivastava N., Hinton G., Krizhevsky A. et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*. 2014. Vol. 15. P. 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html> (дата звернення: 23.05.2025).
16. Szegedy C., Liu W., Jia Y. et al. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. P. 1–9. DOI: 10.1109/CVPR.2015.7298594.

17. Mnih V., Kavukcuoglu K., Silver D. et al. Human-level control through deep reinforcement learning. *Nature*. 2015. Vol. 518, No. 7540. P. 529–533. DOI: 10.1038/nature14236.
18. Hochreiter S., Schmidhuber J. Long short-term memory // *Neural Computation*. 2019. Vol. 9, No. 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
19. Graves A., Mohamed A. R., Hinton G. Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013. P. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.
20. Cho K., Van Merriënboer B., Gulcehre C. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014. P. 1724–1734. DOI: 10.3115/v1/D14-1179.
21. Bengio Y., Courville A., Vincent P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013. Vol. 35, No. 8. P. 1798–1828. DOI: 10.1109/TPAMI.2013.50.
22. Silver D., Schrittwieser J., Simonyan K. et al. Mastering the game of Go without human knowledge. *Nature*. 2017. Vol. 550. P. 354–359. DOI: 10.1038/nature24270.
23. Zhang C., Bengio S., Hardt M., Recht B., Vinyals O. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*. 2017. URL: <https://openreview.net/forum?id=Sy8gdB9xx> (дата звернення: 26.05.2025)
24. Yin W., Kann K., Yu M., Schütze H. Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*. 2017. URL: <https://arxiv.org/abs/1702.01923> (дата звернення: 29.05.2025).
25. Bengio Y. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*. Vol 7700. Berlin, Heidelberg : Springer, 2012. P. 437–478. DOI: 10.1007/978-3-642-35289-8\_26.

26. Wang Y., Zhao L. et al. Evolutionary Strategies for Neural Network Training. IEEE Transactions on Neural Networks and Learning Systems. 2019. Vol. 30, No. 5. P. 1422–1432. DOI: 10.1109/TNNLS.2018.2871698.
27. Yu F., Koltun V. Multi-scale context aggregation by dilated convolutions. International Conference on Learning Representations (ICLR). 2016. arXiv:1511.07122. URL: <https://arxiv.org/abs/1511.07122> (дата звернення: 30.05.2025).
28. Kingma D. P., Welling M. Auto-encoding variational Bayes. International Conference on Learning Representations (ICLR). 2014. arXiv:1312.6114. URL: <https://arxiv.org/abs/1312.6114> (дата звернення: 01.06.2025).
29. Zhang Y., Yang Q. A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering. 2022. Vol. 34, No. 12. P. 5586–5609. DOI: 10.1109/TKDE.2021.3070203.
30. Stanley K. O., Miikkulainen R. Evolving neural networks through augmenting topologies // *Evolutionary Computation*. 2022. Vol. 10, No. 2. P. 99–127. DOI: 10.1162/106365602320169811.