

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Факультет радіофізики, біомедичної електроніки та комп'ютерних систем

«До захисту»

В.о. зав. кафедри фізичної і біомедичної електроніки та комплексних інформаційних технологій

док. фіз.-мат. наук, доцент

_____ С.Л. Бердник

«__» _____ 2024 р

ДИПЛОМНА РОБОТА МАГІСТРА

**МЕТОД РОЗПІЗНАВАННЯ ЖЕСТІВ ДЛЯ БІОНІЧНОГО ПРОТЕЗУ
ВЕРХНЬОЇ КІНЦІВКИ**

Науковий керівник
Кандидат технічних наук,
доцент кафедри біомедичних
електронних пристроїв і систем
ім. В.Н. Каразіна МОН України

_____ О.М. Величко

Виконав студент групи РЕ-61
спеціальність 176 «Мікро- та наносистемна техніка»

_____  М.О. Гриценко

АНОТАЦІЯ

Гриценко Михайло. – Метод розпізнавання жестів для біонічного протезу верхньої кінцівки. – рукопис. Дипломна робота магістра за спеціальністю 153 «Мікро- та наносистемна техніка» – Харківський національний університет ім. В. Н. Каразіна, Харків, 2023.

58 с., 12 рис., 3 табл., 52 джерела

В роботі наведений аналіз технологій заміщення втраченої верхньої кінцівки штучним аналогом – протезом. На основі опрацьованої літератури, були виділені особливості побудови протезів, які можна розділити на пасивні з мінімальним набором функцій, протези, що працюють від тіла на жгутовій системі, електричні протези з власним джерелом живлення та гібридні протези. Електричні протези з власним джерелом живлення, які можуть аналізувати сигнали м'язової (електроміографія), нервової (підшкірні імплантовані електроди) або мозкової активності (електроенцефалографія) задля відтворення намірів людини по виконанню певних дій (біонічні протези) – є областю, яка досліджується найбільше. Проаналізована проблематика області: серед задач, які стоять перед науковою спільнотою можна виділити задачі реалізації протезів плавного руху з великою кількістю ступенів свободи пальців, кісті та плеча, зворотній зв'язок через нервову систему людини та питання розпізнавання біосигналів організму людини задля визначення жесту, який виконується протезом. Для вирішення цього питання класифікації, в роботі був досліджений підхід з використанням машинного навчання. Проаналізовані сигнали ЕМГ 43 учасників, які виконували 17 різноманітних жестів. Для аналізу цих сигналів був власноруч створений набір математичних характеристик сигналу («особливостей» сигналу), який аналізувався за допомогою нейронної мережі глибокого навчання. Результати демонструють, що проста нейронна мережа має змогу розрізняти жести з задовільною (75% точністю) та може використовуватися як базова програма протезу перед коротким періодом донавчання для потреб конкретної людини або як програма для навчання з нуля для індивідуальної програми

SUMMARY

Hrytsenko Mykhailo. – Gesture Recognition Method for Bionic Upper Limb Prosthesis - manuscript. Master's thesis in specialty 153 " Micro- and nanosystems engineering " - V. N. Karazin Kharkiv National University, Kharkiv, 2023.

58 p., 12 figures, 3 tables, 52 sources

The technologies for replacing a lost upper limb with an artificial analog – a prosthesis – are analyzed in this paper. The specifics of prosthesis construction, identified based on the reviewed literature, are categorized into passive prostheses with a minimal set of functions, body-powered prostheses utilizing a harness system, electric prostheses equipped with their own power source, and hybrid prostheses. Among these, electric prostheses with independent power sources, capable of analyzing signals from muscles (electromyography), nerves (subcutaneous implanted electrodes), or brain activity (electroencephalography) to replicate a person's intended actions (bionic prostheses), are recognized as the most researched area. The problems faced by the field are also examined. These include challenges such as implementing smooth movement in prostheses with a high degree of freedom for fingers, hands, and shoulders, providing feedback through the human nervous system, and improving the recognition of biosignals from the human body to determine gestures performed by the prosthesis. To address the classification of biosignals, a Machine Learning approach was investigated. In this study, EMG signals collected from 43 participants performing 17 different gestures were analyzed. To process these signals, a set of mathematical signal characteristics ("signal features") was generated and evaluated using a deep learning neural network. The results indicate that gestures can be distinguished with satisfactory accuracy (75%) using a simple neural network. This network can serve as a foundational prosthetic program, either for initial use before personalized training or as a starting point for creating a customized program tailored to an individual's needs.

ЗМІСТ

1.	<u>Вступ</u>	5
2.	<u>Постановка задачі</u>	13
2.1.	<u>Проблематика побудови біокерованих протезів</u>	13
2.2.	<u>Біосигнали в системі протез-людина</u>	15
3.	<u>Методи та матеріали</u>	21
3.1.	<u>Дані для обробки</u>	21
3.2.	<u>Попередня обробка даних</u>	25
3.3.	<u>Нейронні мережи</u>	31
3.4.	<u>Модель Deep Learning для розпізнавання жестів</u>	41
4.	<u>Результати та обговорення</u>	47
5.	<u>Висновки</u>	51
	<u>Список використаних джерел</u>	52

1. ВСТУП

Людський організм – унікальна і дуже складна система, яка повинна ідеально функціонувати і працювати майже бездоганно протягом усього життя людини. Зокрема, рука є універсальним органом для взаємодії з навколишнім середовищем, який дозволяє захоплювати і носити різні предмети і завдяки рецепторам є джерелом складних тактильних відчуттів, допомагає з балансом тіла та орієнтацією у просторі. Спритність і універсальність руки як маніпулятора чудові та не мають багато аналогів. Ми можемо використовувати наші руки не тільки для того, щоб робити різноманітні жести та точно маніпулювати дуже дрібними об'єктами, але й щоб виконувати складні розумові дії пов'язані з рухом, наприклад, грати на музичних інструментах або працювати зі побутовими технічними пристроями. Втрата руки різко погіршує якість життя людини, оскільки прості дії, такі як застібання гудзика чи зав'язування шнурка, стають проблемою. А втрата обох рук робить практично неможливим та невиносним самостійне, незалежне існування і прирікає людину на постійну опіку. За даними Всесвітньої організації охорони здоров'я, станом на зараз, приблизно 1,3 мільярда людей (16% населення світу) мають постійну інвалідність [1], з яких щонайменше 10 мільйонів - це люди з ампутацією; з них 3 мільйони людей мають ампутацію верхньої кінцівки. Міжнародні дослідження щодо причин та поширеності таких ампутацій обмежені. Основною причиною є травма, частіше за все внаслідок нещасних випадків на виробництві, дорожньо-транспортних пригод та падінь. Повернення навіть замінної руки таким людям значно підвищує якість життя людей. Як рішення цієї проблеми, можна виділити трансплантацію та протезування.

Трансплантація руки має значні обмеження через складну та дорогую процедуру операції, труднощі з пошуком відповідного донора та необхідність використання імуносупресивні препарати протягом усього життя пацієнта [2]. Незважаючи на те, що трансплантологія рук зазнала значного прогресу, вона все ще не має широкого медичного поширення порівняно із історично відомою

людству заміною втраченої кінцівки на штучний аналог. Одна з перших згадок про протез руки у вигляді металевої репліки була описана ще в 77 році н.е. римським вченим Плінієм Старшим у його енциклопедії «Природнича історія» [3] і з того часу процес та технологія заміщення верхніх кінцівок пройшли значний шлях вдосконалення. На сьогоднішній день, протези можна розділити за типом енергопостачання, програмою керування, набором функцій та рівнем ампутації.

Ампутація однієї кінцівки називається «одностороння ампутація», тоді як ампутація і правої, і лівої кінцівки називається «двостороння ампутація». Враховуючи додаткові труднощі у виконанні завдань при двосторонній ампутації, протези, що використовуються при двосторонній ампутації, часто мають додаткові функції [4]. Ампутації верхніх кінцівок відрізняються за рівнем (рис.1.1). Хоча вроджену відсутність кінцівки класифікують за дещо зміненою таксономією, люди можуть користуватися протезами на відповідному рівні ампутації, незалежно від того, чи є дефект кінцівки вродженим або набутиим.

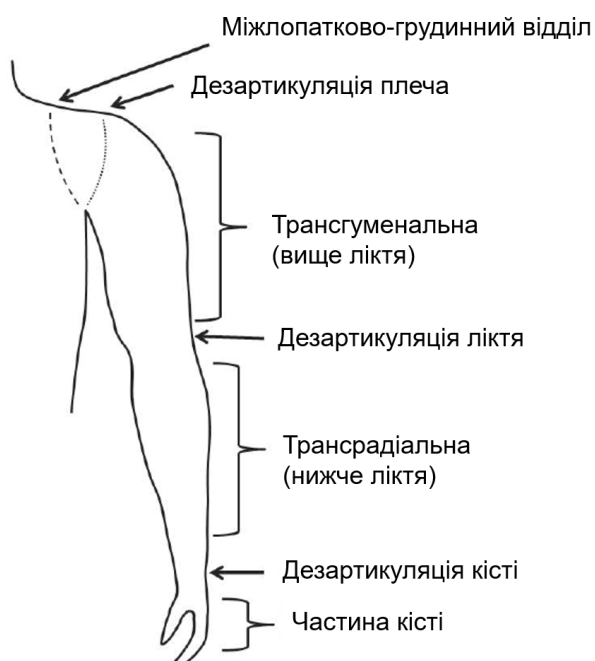


Рисунок 1.1 – Рівні ампутації верхньої кінцівки

Частина кінцівки, що залишилася, часто називають «кукса». «Трансрадіальна ампутація» (або “ампуація нижче ліктя”) відбувається через

довгі кістки променевої та ліктьової, “трансплечова ампутація” (або “ампутація вище ліктя”) відбувається через плечову кістку. Ампутація через суглоб називається «дизартикуляцією». Таким чином, людині може бути виконана дизартикуляційна ампутація зап'ястя (між зап'ястком і променевою та ліктьовою кістками), дизартикуляційна ампутація ліктя (між променевою та ліктьовою кістками і плечовою кісткою) або дизартикуляційна ампутація плеча (між плечовою кісткою і лопаткою).

Ампутації проксимально (ближче до центру тіла) до плеча називаються «міжлопатково-грудними ампутаціями» (іноді їх ще називають «ампутаціями передньої чверті»). Часткові ампутації кисті - це ампутації дистальніше променево-зап'ясткового суглоба, які найкраще описуються за допомогою термінології Міжнародної організації зі стандартизації (фалангова, п'ястково-фалангова, п'ясткова, п'ястково-зап'ясткова та зап'ясткова). Також вони мають назву транспаральної ампутації [5].

Кількість ампутацій різних рівнів відрізняється (рис. 1.2). 61% - транскарпальна ампутація, 16% становить транс-плечова та 12% - трансрадіальна, що робить їх найбільш поширеними видами втрати кінцівки, що прямопорційно кількості протезів, що розробляються; інші види мають наступний розподіл: 2% - передня чверть, 3% - дисартикуляція плеча, 1% - дисартикуляція ліктьового суглоба, 2% - дисартикуляція зап'ястя, 3% - двостороння втрата кінцівки [6].

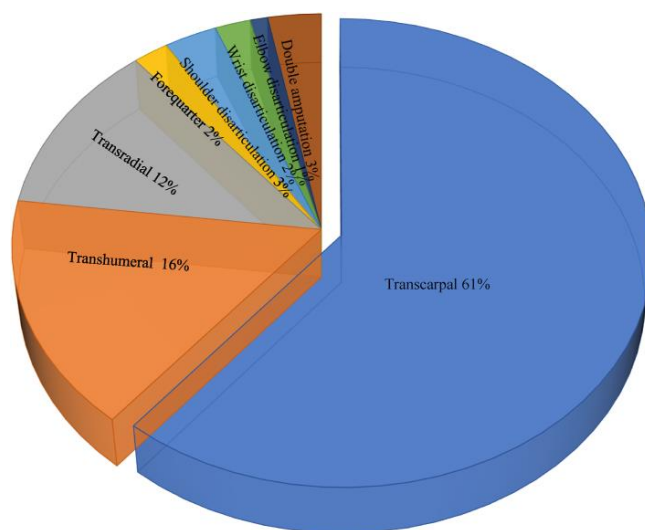


Рисунок. 1.2 – Розподіл рівнів ампутації кінцівок

Звичайно, що відповідні рівні ампутації потребують відповідного технічного рішення щодо заміщення втрачених функцій, в першу чергу складних функцій кисті, яка статистично є найбільш розповсюдженим рівнем ампутації. Перше, що потребує уваги – це джерело енергії для протеза та спосіб перетворення цієї енергії у механічний рух. Серед видів енергопостачання можна виділити 4 основних типів протезів верхніх кінцівок. 1 Пасивні протези (рис. 1.3) відновлюють антропоморфну довжину та масу кінцівки, необхідну для перенесення або стабілізації предмета під час іпсилатерального або білатерального використання, але активно не рухаються. Пасивний пристрій може також забезпечувати опору для решти пальців, як у випадку з людьми з частковою презентацією кисті. Пасивні протези можна описати як статичні або позиційні. Статичні пристрої не мають рухомих



Рисунок 1.3 – Пасивний протез

частин, в той час як позиційні конструкції включають гнучку арматуру або з'єднання з храповим механізмом. Статичні пасивні пристрої не забезпечують активного захвату, тому користувач має відносно обмежену здатність до захвату порівняно з позиційними пасивними пристроями. Цей варіант, як правило, створюється з урахуванням індивідуальних завдань і для найбільш складних зовнішніх умов, де інші варіанти мають більше шансів вийти з ладу. Ці протези виглядають як природна кінцівка, є найлегшими у виробництві та

найдешевшими для пацієнта, але не забезпечують активних рухів рукою та суглобами [7].

2. У протезах, що працюють від тіла (рис. 1.4), використовується джгут (або система джгутів), щоб використовувати проксимальний рух тіла, що викликає рух кабелю. Кабель закінчується на гачку або руці і впливає на рух механічних компонентів [8]. Протези, що рухаються від тіла, мають ряд переваг перед альтернативними варіантами. Зазвичай вони менші за вагою, міцніші, більш стійкі до умов навколишнього середовища (наприклад, вологи, пилу тощо), забезпечують вторинний пропріоцептивний зворотний зв'язок з користувачем через напругу джгута, мають нижчу початкову ціну та менші витрати на технічне обслуговування порівняно з до електричних варіантів [9].



Рисунок 1.4 – Протез, що працює від тіла та тросова система

Незважаючи на те, що ці пристрої вважаються більш довговічними та міцними, а люди, які займаються фізичною працею, зазвичай віддають перевагу цьому типу, очікується, що при інтенсивному використанні ремонт джготових систем, кінцевих пристроїв та інших компонентів буде частим і необхідним для підтримки працездатності людини. Не дивлячись на переваги

протезів, що працюють від енергії рухів тіла, є ряд недоліків: джгути обмежують пацієнта, користувачі з більш проксимальними рівнями різниці кінцівок мають більші труднощі з генеруванням необхідного зміщення для роботи цих пристроїв, а деяким особам не подобається зовнішній вигляд гачків, кабелю та джгутів. Розробки таких протезів хоча і продовжуються [10], удосконаленню існуючих систем не приділяється багато уваги [11].

3. Протези з електричним приводом (рис. 1.5) використовують двигуни для здійснення руху, які живляться від системи акумуляторних батарей. Найпоширеніший метод керування - за допомогою сигналів електроміографії (ЕМГ) або електроенцефалографії (ЕЕГ). Такі протези включають в себе сервоприводи, які забезпечують більшу силу захоплення, ніж протези з живленням від тіла [12], лінійні потенціометри або датчики, резистори для вимірювання сили, кулісні перемикачі, кнопкові перемикачі та перемикачі джгута, а також інерційні вимірювальні пристрої. Існує також можливість використання декількох схем управління в одному протезі.



Рисунок 1.5 – Протези з електричним приводом

Варіанти протезних компонентів з електричним приводом варіюються від кисті та зап'ястя до ліктя та плеча. Переваги протезів з електричним приводом включають швидкість і силу стискання, можливість регулювання залежно від сили м'язового скорочення, більш антропоморфний зовнішній

вигляд, можливість збільшення функціонального діапазону роботи. До недоліків цього протеза можна віднести суттєво вищу початкову вартість (до 100 000 долларів), необхідність обслуговування батареї, більшу вагу, складніший ремонт, непереносимість вологого, брудного або агресивного середовища.

4. Гібридні протези (рис. 1.6) це об'єднання останніх двох вищеописаних варіантів протезування в одну біотехнологічну систему. Найпоширеніша гібридна конфігурація - це лікоть з електроприводом, який керується ременева системою на тілі, та кисть/зап'ястя з електроприводом для пацієнтів з ампутацією вище ліктя [13], керування якою може бути реалізовано різними способами. Переваги та недоліки гібридного протеза залежать від того, які компоненти живляться і яким чином. Як правило, гібридні пристрої забезпечують одночасний контроль над декількома компонентами, мають



Рисунок 1.6 – Гібридний протез

меншу вагу, ніж повністю електричні протези, і пропонують більшу силу захоплення порівняно з протезами, що «живляться» від тіла. Недоліки, як зазначено вище, полягають у тому, що ліктьовий протез з електроприводом повинен рухатися під дією сили тяжіння або ременя і не може забезпечувати

підйом за допомогою сили тяжіння. Необхідність використання ременів для керування компонентами, що приводяться в рух власним тілом, має той же самий ряд недоліків, що описаний вище для жгутової системи.

2. ПОСТАНОВКА ЗАДАЧІ

2.1 Проблематика побудови біокерованих протезів

Функції верхніх кінцівок набагато складніше замінити, ніж функції нижніх кінцівок. Це пов'язано з тим, що істотно нижні кінцівки більш обмежені і використовуються, в першу чергу, для підтримання та досягнення вертикального положення і різних видів локомоції (наприклад, ходьба, біг, стрибки, підстрибування, підйом по сходах). На відміну від цього, основні функції верхніх кінцівок включають не тільки грубу і дрібну моторику, а й більш складні комбінації видів діяльності, такі як самообслуговування, взаємодія з навколишнім середовищем та іншими людьми і самовираження. Верхня кінцівка - це система з високим ступенем свободи, що дозволяє рухати рукою в різних положеннях навколо тіла. Складність верхньої кінцівки людини ілюструється великою часткою простору в моторній та сенсорній ділянках мозку (моторні та сенсорні гомункули), що відповідають за руховий контроль та відчуття кисті та пальців. Тому не дивно, що люди з ампутацією верхньої кінцівки загалом менш задоволені відновленням функцій, які забезпечують їхні протези, порівняно з людьми з ампутацією нижньої кінцівки. Відповідно до [14], результати детального анкетування людей з ампутуваними верхніми кінцівками з приводу того, як часто вони носять свої протези та наскільки вони задоволені своїми протезами та їх функціональними можливостям, 56% ампутантів носили свої протези «час від часу» або «ніколи». Протези найчастіше носили «постійно» для роботи та соціальної активності. Кількість часу, протягом якого люди з ампутуваними кінцівками носили протези, була помірно пов'язана з рівнем їхньої задоволеності протезами. 64% ампутуваних оцінили свої протези як «задовільні» або «неприйнятні». Це пов'язано, як і з психологічним станом пацієнта [15] та соціальним сьогоденням з переважно негативним або «особливим» ставленням суспільства до людей з протезами кінцівок [16] (до 74% опитуваних не приймали людей з протезами незалежно від віку, статі та

освіти), так і з якістю протезів, які не завжди в стані забезпечити навіть мінімальні повсякденні потреби.

На сьогоднішній день, найбільш перспективним напрямком є розробка повністю автономних у живленні, біонічних (тобто таких, що відтворюють природні можливості) протезів [17]. Розробка таких пристроїв має багато труднощів, зокрема проблеми побудови суглобів та високофункціональних та високошвидкісних елементів кісті (рис. 2.1). Традиційно активні протези зап'ястя забезпечують лише пронацію/супінацію (так звану ротацією зап'ястя), хоча положення зап'ястя в інших двох ступенях свободи також важливо для полегшення захоплення об'єкта та маніпуляції. Цікаве дослідження показало, що наявність більшої кількості степенів свободи на зап'ясті була такою ж важливою, як (або важливіша, ніж) спритність пальців у конструкції протезної руки [18]. Інше цікаве дослідження [19] в якому запропонували конструкцію, що відтворює вказівний і великий пальці з кісток, зв'язок, сухожилів і м'язового механізму агоніст-антагоніст для протезу, який має таку ж гнучкість та антропоморфний вигляд, як і людська рука.



Рисунок 2.1 – Протез кісті

Технічні проблеми у побудові протезів стосуються не тільки кісті. Хоча ліктьовий суглоб (рис. 2.2) кінематично простіший, ніж суглоби зап'ястя або кісті, він характеризується набагато більшими вимогами до крутного моменту та швидкості порівняно з цими суглобами. Коли рухливість у ліктьовому

суглобі обмежена або відсутня, виконання повсякденних дій за участю верхньої кінцівки стає значно складнішим, і такі дії, як правило, вимагають значних компенсаторних рухів [20]. Окрім характеристик продуктивності, є



Рисунок 2.2 – Побудова типового електричного протезу ліктя
кілька додаткових цілей проектування. Конструкція ліктя повинна точно відповідати антропоморфній оболонці. Важливо, щоб довжина проксимального сегмента ліктьового протеза була мінімізована, а маса підібрана, щоб відповідати потенційним користувачам із довгою залишковою кінцівкою. Крім того, промислово виготовлений дистальний сегмент (передпліччя) має підходити якомога ширшій групі населення, а також залишати місце для використання ротатора зап'ястя, якщо цього бажає користувач.

2.2 Біосигнали в системі людина-протез

Окрім чисто інженерних проблем, перед сучасною наукою постає питання зчитування, аналізу та категоризації сигналів що, відповідають різним жестам, а також зворотній зв'язок для відтворення почуттів. Останніми роками багато досліджень зосереджені на вилученні корисної інформації з біологічних сигналів з метою належного контролю протезів верхніх кінцівок. Традиційно, поверхнева ЕМГ (пЕМГ), яка зображена на рис. 2.3, є найбільш поширеним сигналом для контролю протезів, особливо для контролю кисті

руки, яка є найбільш часто ампутованою частиною, але її використання все ще стикається з багатьма недоліками [21].

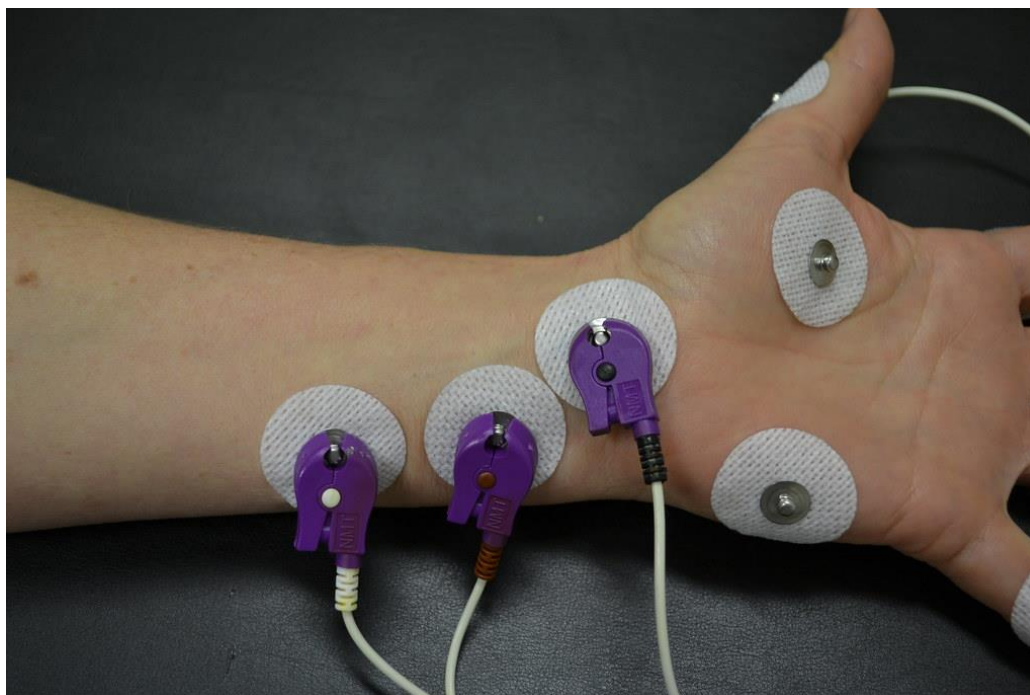


Рисунок 2.3 – Електроди поверхневої електроміограми

У той час як косметика протезування, електронні компоненти та обчислювальні можливості зазнали значного вдосконалення, стратегії контролю, які зараз використовуються в протезуванні, не змінилися з моменту їхньої першої появи в 1960-х роках. Тож, пЕМГ є одним з основних джерел для контролю протезів. Це метод вивчення активації скелетних м'язів шляхом реєстрації електричних потенціалів, що виникають при скороченні цих м'язів. Теорія, що лежить в основі електродів пЕМГ, полягає в тому, що вони утворюють хімічну рівновагу між детектуючою поверхнею електрода і шкірою тіла завдяки електролітичній провідності, завдяки чому струм може протікати в електрод. Для отримання бажаного жесту з оброблених пЕМГ-сигналів використовується кілька методів, які використовують той факт, що люди з ампутованими кінцівками все ще можуть генерувати різні і повторювані м'язові патерни, пов'язані з різними рухами передпліччя за допомогою залишкових м'язів культі. ЕМГ низької щільності (з меншою кількістю контактів або менш щільним розташуванням) широко

використовується в протезуванні, як в дослідницькому, так і в комерційному контексті. ЕМГ можна класифікувати відповідно до рівня роздільної здатності та щільності датчиків. пЕМГ можна класифікувати за кількістю використовуваних електродів. ЕМГ низької щільності зазвичай стосується використання невеликої (<10) кількості біполярних датчиків ЕМГ, які можуть бути або вологими, тобто містити електролітичну речовину, яка служить як розділ між шкірою та електродами, або сухими. І навпаки, ЕМГ високої щільності зазвичай складається з вологих монополярних датчиків, розташованих на ділянці руки, приблизно на відстані 1 см один від одного, і з опорним заземленням, як правило, розміщеним на зап'ясті або на лікті. Важливо, що електроди пЕМГ також відрізняються за своєю електронною конфігурацією, оскільки вони можуть бути або попередньо підсилені, або ні. Протезний контроль за допомогою ЕМГ низької щільності зазвичай досягається за допомогою двох біполярних електродів, розміщених на м'язах-антагоністах. Ця конфігурація дозволяє контролювати протезну систему надійним і простим способом, який не потребує значних енергетичних втрат, але не завжди має достатню роздільну здатність для розрізнення великої кількості рухів. Проте виявлення складних і одночасних рухів фантомної кінцівки можна покращити за допомогою масиву електродів ЕМГ, розміщених на поверхневій шкірі залишкового передпліччя [22- 24]. При побудові електроміографічної системи аналізу сигналу, треба враховувати вплив підвищення щільності електродів на електроспоживання. Варто зазначити, що ЕМГ-сигнали також можуть бути зібрані за допомогою інвазивних методів.

Також, серед методів збору даних можна виділити електроенцефалограму (ЕЕГ), під час якої зчитуються значення електропотенціалів мозку людини для подальшого використання в класифікації жестів. Цей напрямок активно розвивається, але має наступний ряд недоліків. Електричний потенціал спадає оберненопропорційно квадрату відстані, тому поверхнева ЕЕГ має занадто низьку роздільну здатність для складної задачі мультикласифікації. Це призводить до того, що кількість

жестів, які може реалізовувати ЕЕГ-керований протез або низька [25] та дорівнює 1-3 жестах, або потребує специфічних розумових зусиль, як то зосередження або розслаблення мозку протягом певної кількості часу [26].

Варто згадати й зворотній зв'язок, як новітню проблему у побудові протезів. Нерви у верхній кінцівці, що беруть початок від нервових корінців С5 до Т1, просуваючись дистально, об'єднуються у пучки, які відгалужуються від нервового пучка для еферентної іннервації м'язів та аферентної іннервації шкіри, сухожилів і кісток. Існує багато відмінностей в аксонах периферичних нервів (наприклад, мієлінізація, діаметр, швидкість проведення), які відповідають цільовому об'єкту іннервації, наприклад, руховій одиниці в м'язах або різним рецепторам сенсорної нервової системи. Ці сенсорні рецептори класифікуються як механорецептори, які відчують дотик і тиск, терморецептори, які відчують зміни температури, і ноцицептори, які відчують біль [27]. Оптимальний сенсорний контур зворотного зв'язку повинен складатися з механічних рецепторів на поверхні протеза, які можуть перетворювати зовнішні подразники на електричні сигнали. У системі з імплантованими нейронними електродами ці закодовані сигнали стимуляції можна було б використовувати для стимуляції окремих ділянок змішаного нервового пучка, які передають сенсорну інформацію до відповідної частини мозку. Ідеальний стимулятор повинен мати як високу селективність за типом волокон, контролюючи тип стимулів, що сприймаються (постійний тиск, вібрація, тепло), націлюючись на певні групи волокон (як повільно адаптуються, швидко адаптуються, або взагалі є вільними нервовими закінчення), так і високу просторову селективність, зменшуючи кількість індукованих потенціалів дії, що не відносяться до цілі, що забезпечує вищу роздільну здатність стимуляції. Покращення узгодження місцезнаходження та швидкості призводить до зменшення часу, який пацієнти витрачають на тренування для інтерпретації сигналів, які вони отримують від протезу. Це хірургічний та інженерний виклик на рівні нейронного інтерфейсу, який не

можна перекласти на здатність нервової системи змінювати сприйняту локалізацію. [28]

Тим не менш, основною проблемою на сьогодні є суттєво обмежена кількість рухів, які може виконувати протез та швидкість виконання цих рухів, що в першу чергу впливає на задоволення пристроєм. Після отримання сигналів пЕМГ існує відома проблема інтерпретації сигналів та їхньої відповідності певним жестам у реальному часі. Ця проблема може бути вирішена на основі логічних схем категоризації, наприклад, так званої контекст-залежної класифікації [29], коли пацієнт може відтворювати певну кількість повністю контрольованих скорочень м'язів куксою, кожному з яких надається певний клас руху протеза, який в свою чергу використовується для ініціації відкриття наступного шару класифікатора, в якому ті ж самі сигнали скорочень м'язів набувають новий контекст. Хоча це рішення є менш інтуїтивним порівняно з класичною схемою, у якій кожен клас сигналів пЕМГ завжди відповідає одному руху протеза, воно має практичні властивості та високий потенціал для застосування. Оскільки інтуїтивність є питанням розвитку уявлення користувача, належна методика тренування може прискорити момент вже практичного використання. Тим не менш ця схема потребує довгої роботи пацієнта зі спеціалістом і постійної уваги, через те, що рухи протеза не відповідають м'язовим скороченням, тим паче, якщо потрібно запам'ятовувати гілки контекстних класифікацій. Через це, виникає потреба в інших способах аналізу та класифікації жести. Гарним вирішенням цього питання стали різноманітні застосування машинного навчання [30], з використанням різноманітних моделей, та обробки даних, починаючи від аналізу спектрограм сигналів [31] до знаходження специфічних характеристик міографічних сигналів [32]. Тим не менш, багато з цих моделей або обмежені малою кількістю жестів (5-7) та різними оцінками точності класифікації, або потребують дуже довгого навчання з оператором, що робить коштовне протезування ще дорожчим. Тим не менш, нейронні мережі є найбільш

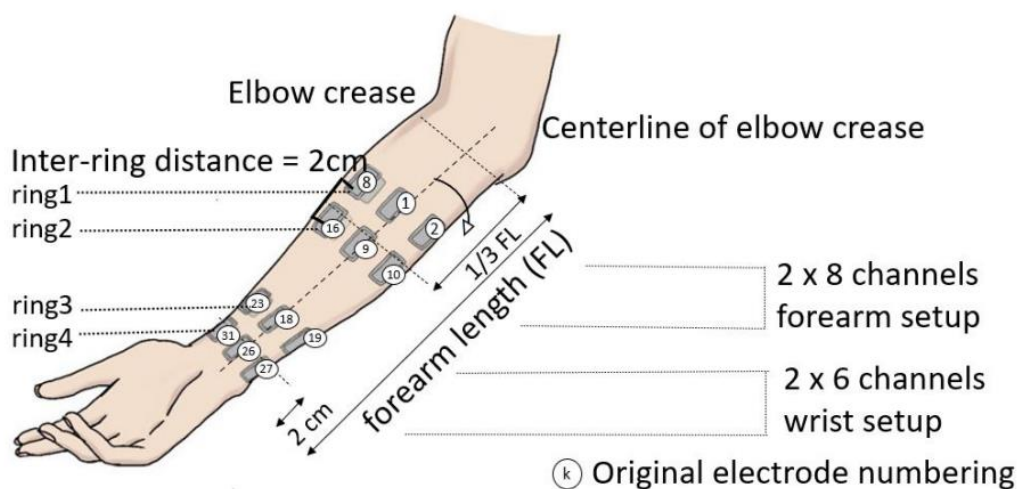
зручним для використання класифікатором, завдяки автоматичному прийняттю рішень без додаткових зусиль з боку ампутанта.

На основі проведеного аналізу, виділимо проблему розпізнавання біосигналів для класифікації жестів. Мета цієї роботи – попередній аналіз електроміограми м'язових скорочень кінцівки пацієнтів (як найбільш зручного методу збору даних та керування протезом) для великої кількості жестів та розробка моделі простої нейронної мережі для класифікації цих жестів, яка може бути використана як базова при швидкому донавчанні готового протеза на певних пацієнтах, що робить технологію більш доступною або для повного навчання для конкретної людини.

3. МЕТОДИ ТА МАТЕРІАЛИ

3.1 Данні для обробки

Як набір даних для аналізу, використовувалися данні міографічного дослідження [33]. У дослідженні взяли участь 43 здорових учасники (23 чоловіки, 20 жінок). Середній вік становив $26,35 \pm 2,89$ років, а середня довжина передпліччя (від стилоїдного відростка зап'ястя до олекранона ліктя) становила $25,15 \pm 1,74$ см. Експерименти проводилися відповідно до Декларації Гельсінкі, а протокол дослідження був схвалений Етичним комітетом Університету Ватерлоо. Перед початком експерименту довжина передпліччя вимірювалася як відстань між олекранонним відростком і ліктьовим стилоїдним відростком. Окружність передпліччя вимірювалася на відстані однієї третини довжини передпліччя від олекранонного відростка. Окружність зап'ястя вимірювалася на відстані 2 см від ліктьового стилоїдного відростка. Після проведення цих вимірювань електроди встановлювалися на передпліччя та зап'ястя (див. рис. 3.1). Для розміщення електродів на передпліччі використовували шістнадцять однополюсних пЕМГ-електродів (AM-N00S/E, Ambu, Данія), які розташовувалися у вигляді двох кілець, кожне з яких складалося з восьми електродів, рівномірно розташованих навколо передпліччя і формуючи вісім біполярних пар. Відстань між центрами двох кілець становила 2 см.



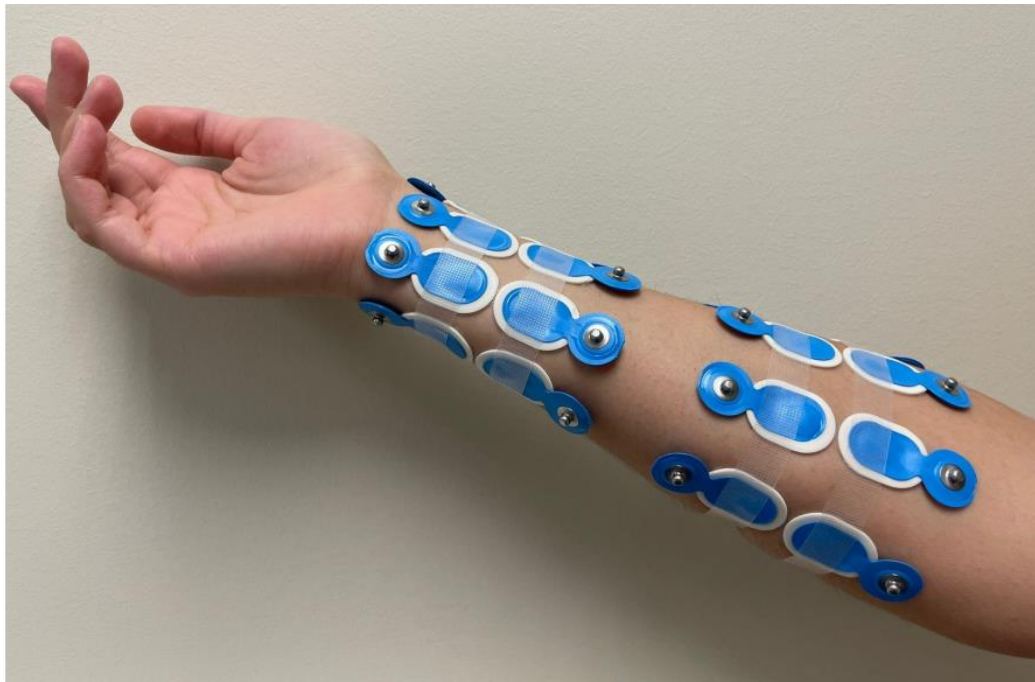


Рисунок 3.1 – Розташування електроміографічних електродів

Для встановлення електродів на зап'ясті використовували дванадцять однополюсних пЕМГ-електродів такого ж типу, які розташовувалися у вигляді двох кілець, кожне з яких складалося з шести електродів, рівномірно розташованих навколо зап'ястя і формуючи шість біполярних пар. Відстань між центрами кілець також становила 2 см, як і у випадку передпліччя. Таким чином, для кожної сесії використовувалося загалом 28 однополюсних пЕМГ-електродів. Таке розташування електродів ще не вважається повноцінно щільним, але вже дає достатню, як буде показано далі, кількість інформації для задачі класифікації. Щоб забезпечити однакове розташування електродів у всіх учасників, перший електрод у кожному кільці (загалом 4 кільця) анатомічно розташовувався на центральній лінії згину ліктя. Після налаштування учасник сідав зручно на стілець із обома верхніми кінцівками в розслабленій позиції. Учасники виконували 16 жестів рук (рис. 3.2) і зап'ястя з використанням

TABLE
GESTURE LIST













Gesture	Description	Gesture	Description
	Lateral prehension (LP)		Index finger extension (IFE)
	Thumb adduction (TA)		Thumb extension (TE)
	Thumb and little finger opposition (TLFO)		Wrist flexion (WF)
	Thumb and index finger opposition (TIFO)		Wrist extension (WE)
	Thumb and little finger extension (TLFE)		Forearm pronation (FP)
	Thumb and index finger extension (TIFE)		Forearm supination (FS)
	Index and middle finger extension (IMFE)		Hand open (HO)
	Little finger extension (LFE)		Hand close (HC)

Рисунок 3.2 – Набір з 17 жестів, що досліджувалися візуальних інструкцій на екрані комп'ютера. Дані про стан спокою збиралися після кожного випробування, яке включало 16 жестів (тобто загалом 17 жестів разом зі спокоєм). Кожен жест повторювався сім разів (17 жестів \times 7 випробувань = 119 скорочень) з п'ятисекундним записом у кожному випробуванні. Для уникнення м'язової втоми між послідовними скороченнями надавався десятисекундний період відпочинку.

Для кожного 5-секундного запису ЕМГ (рис. 3.3) існує 32 сигнали/канали, які записуються з частотою 2048 Гц (розмір матриці: 10240 \times 32). З цих 32 каналів 16 каналів збираються з передпліччя, а 12 каналів — із зап'ястя у біполярній конфігурації (8+8 для передпліччя і 6+6 для зап'ястя у формі 4 кілець), тоді як 4 канали залишаються невикористаними.

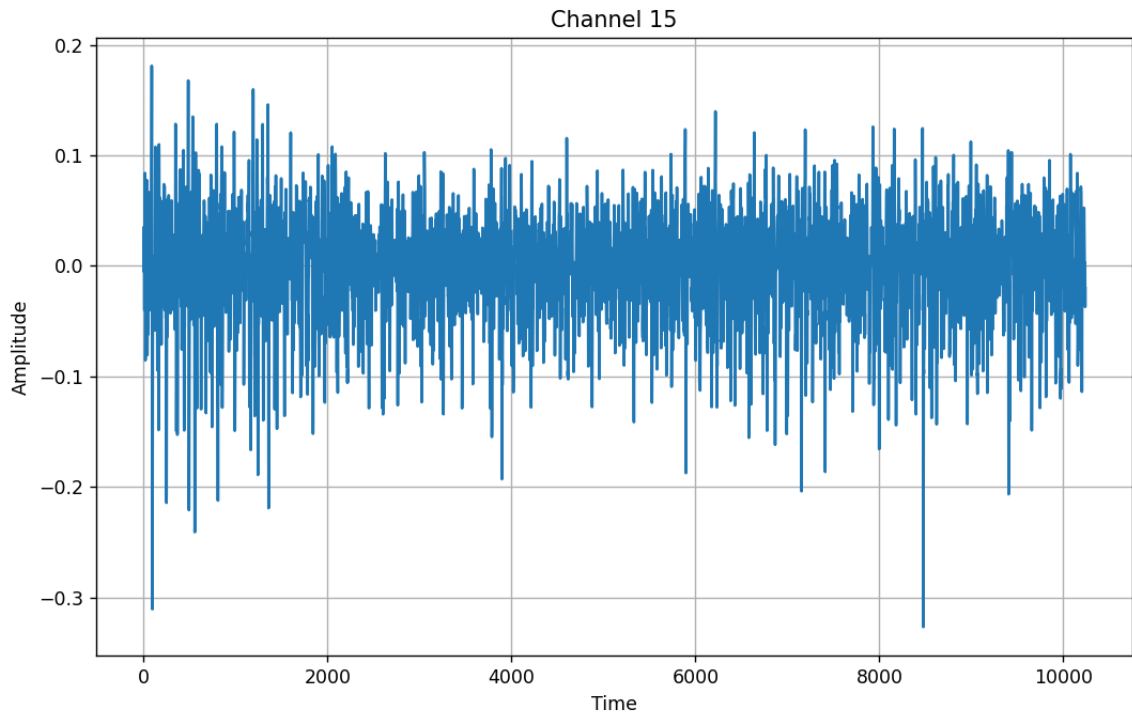


Рисунок 3.3 – Типовий 5-секундний сигнал ЕМГ

Таким чином, дані містять 17 жестів по 7 випробувань на жест, для 43 учасників протягом 3 сесій, тобто 903 сигнали на жест, кожний розміром 10240 відліків на 28 каналів (враховуючи невикористаність 4 каналів). Дані дослідження розподілені наступним чином: три верхньорівневі папки представляють три сесії. В кожній папці сесії є 43 підпапки для кожного з учасників. Три папки для трьох сесій збору даних: "Session1", "Session2" і "Session3". Кожна з цих папок містить підпапки з назвами "sessioni_subjectj" для кожного учасника. Кожна підпапка учасника містить файли записів даних, назви яких мають формат "sessioni_subjectj_gesturek_trialt.dat" та "sessioni_subjectj_gesturek_trialt.he". У цих файлах, $i \in \{1,2,3\}$ — це індекси сесій, $j \in \{1,2,\dots,43\}$ — індекси учасників, $k \in \{1,2,\dots,16\}$ — індекси жестів, $t \in \{1,2,\dots,7\}$ — індекси випробувань. Канали передпліччя перераховані як {F1-F8} (відповідають кільцю 1 з номерами стовпців {1,2,...8}) і {F9-F16} (відповідають кільцю 2 з номерами стовпців {9,10,...16}). Канали зап'ястя пронумеровані як {W1-W6} (відповідають кільцю 3 з номерами стовпців {18,19,...23}) і {W7-W12} (відповідають кільцю 4 з номерами стовпців {26,27,...31}).

Для перетворення даних на формат .mat, з яким бібліотеки Python легко працює, використовується бібліотека *wfdb*.

3.2 Попередня обробка даних

Перетворені на .mat для кожного учасника файли були об'єднані у єдину послідовність, яка містить $3 \times 17 \times 7 = 357$ матриць 10240×28 . Через те, що наша робота присвячена аналізу за допомогою Deep Learning методики, яка буде широко описана в наступному розділі, кількість даних для обробки треба значно скоротити, щоб зменшити кількість необхідних множників ваги. Зазвичай, спочатку сигнал, «розрізають» на невеликі проміжки довжиною до 250 мс, як це пропонується у [34, 35] на основі даних, отриманих експериментально. В нашому випадку, кожен сигнал був розділений на 32 проміжки по 320 відліків, що відповідає 156,25 мілісекундам. Така необхідність цілком зрозуміла – на меншій кількості даних може не відстежуватися залежність жесту від сигналу, а більша довжина значить затримку у роботі протезу. Таким чином, для кожного учасника отримано 11424 матриці 320×28 . Кожна з цих матриць є вхідним значенням нейронної мережі, що все ще є достатньо великою кількістю даних для аналізу.

Міоелектричний сигнал утворюється накладанням окремих потенціалів дії, що генеруються нерегулярними викидами активних моторних одиниць у м'язі. Через велику кількість перекритих потенціалів дії моторних одиниць і нерегулярний характер викидів, поверхневий сигнал можна вважати складним і нестабільним випадковим сигналом. Характеристики сигналу сильно залежать від рівня та тривалості скорочення, динамічних або статичних станів м'язів, втоми та потовиділення на шкірі. Тому, замість аналізу самих сигналів, ряд авторів [36, 37] рекомендує перейти до аналізу так званих «особливостей» (features) сигналів, які можна обчислити власноруч заздалегіть і які будуть загальними «гіперхарактеристиками» для розпізнавання сигналів [38, 39]. В цій роботі до кожного з «нарізаних» сигналів вирішено було знаходити наступні характеристики (обрані на основі аналізу літератури), які повернули загалом 22 особливості:

1. Інтеграл ЕМГ та середня енергія сигналу (IEMG and Mean power). Це характеристики, що відображають загальне значення енергії, витраченої на рух.

```
def getIEMG(vector):  
    vector = np.asarray(vector)  
    return np.sum(np.abs(vector))
```

```
def getMNP(vector):  
    vector = np.asarray(vector)  
    spec = np.fft.fft(vector)  
    spec = spec[0:int(round(spec.shape[0]/2))]  
    POW = spec*np.conj(spec)  
    return np.real(np.sum(POW)/POW.shape[0])
```

2. Асиметрія (Skewness) вказує на те, наскільки сигнал є асиметричним навколо середнього значення. Для унімодального розподілу (розподілу з одним піком) від'ємне значення асиметрії зазвичай вказує на те, що хвіст знаходиться ліворуч від розподілу, а позитивний перекик вказує на те, що хвіст знаходиться праворуч:

$$Skewness = \frac{n}{(n-1)(n-2)} \sum_{i=1}^n \left(\frac{x_i - mean}{\sigma} \right)^3$$

Де σ – середньоквадратичне відхилення.

Це можна використовувати для аналізу того, чи була енергія більшою на початку чи в кінці жесту.

```
def getSKEW(vector):  
    vector = np.asarray(vector)  
    return stats.skew(vector)
```

3. Ексцес (Kurtosis) визначає значення крутизни сигналу, тобто наскільки сильно сигнал зосереджений навколо середнього значення:

$$Kurtosis = \frac{\mu_4}{\sigma^4}$$

де μ_4 – четвертий центральний момент розподілу, $\mu_k = E[(X - EX)^k]$

Значення ексцесу зменшується зі збільшенням скорочення м'язів.

```
def getKURT(vector):  
    vector = np.asarray(vector)  
    return stats.kurtosis(vector)
```

4. Середня та медіанна частота. ЕМГ містить різні частоти в різні моменти часу. Одними з найбільш поширених методів для визначення частотного спектра ЕМГ є швидке перетворення Фур'є (FFT) і короткочасне перетворення Фур'є (STFT). Відповідно до [40] ці характеристики потрібно шукати у необроблених даних. Використовуючи частотний спектр, можна розрахувати амплітуду кожної частоти. На основі цих даних можуть бути виділені такі характеристики, як середня частота (MNF), медіанна частота (MDF) тощо. Для отримання репрезентативних характеристик ми маємо як раз достатню кількість вибірок. У результаті, значення MNF і MDF повинні зростати зі збільшенням м'язової сили або навантаження.

```
def getMNF(vector, fs=1000):  
    vector = np.asarray(vector)  
    spec = np.fft.fft(vector)  
    f = np.fft.fftfreq(vector.shape[-1])*fs  
    spec = spec[0:int(round(spec.shape[0]/2))]  
    f = f[0:int(round(f.shape[0]/2))]  
    POW = spec * np.conj(spec)  
  
    return np.real(np.sum(POW*f)/sum(POW))
```

```

def getMDF(vector, fs=1000):
    vector = np.asarray(vector)
    spec = np.fft.fft(vector)
    spec = spec[0:int(round(spec.shape[0]/2))]
    #f = np.fft.fftfreq(vector.shape[-1])
    POW = spec * np.conj(spec)
    totalPOW = np.sum(POW)
    medfreq = 0
    for i in range(0, vector.shape[0]):
        if np.sum(POW[0:i]) > 0.5 * totalPOW:
            medfreq = fs*i/vector.shape[0]
            break
    return np.real(medfreq)

```

5. Перетин нулів (Zero-crossing). Може бути використаний альтернативно або доповнюючи значення середньої частоти з аналогічним зв'язком з рухом м'язів [41].

```

def getZC(vector, threshold=0.1):
    vector = np.asarray(vector)
    number_zero_crossing = 0
    current_sign = cmp(vector[0], b:0)
    for i in range(0, len(vector)):
        if current_sign == -1:
            if current_sign != cmp(vector[i], threshold): # We give a delta to consider that the zero was crossed
                current_sign = cmp(vector[i], b:0)
                number_zero_crossing += 1
        else:
            if current_sign != cmp(vector[i], -threshold):
                current_sign = cmp(vector[i], b:0)
                number_zero_crossing += 1
    return number_zero_crossing

```

6. Розмах сигналу (Range) – може бути використаний для оцінки м'язової сили

```

def getRANGE(vector, filesize=2):
    vector = np.asarray(vector)
    return vector.max()-vector.min()

```

7. Автокореляційні, диференційний автокореляційний коефіцієнти та кепстральні коефіцієнт (Autoregressive coefficient, Difference autoregressive coefficient, Cepstral coefficients) . У [42, 43] доведено експериментально, що ці коефіцієнти прямо впливають на якість розпізнавання. Результати експерименті показали, що як AR-

модельовання, так і кепстральний аналіз є ефективними методами для виявлення аномалій у ЕМГ сигналах. Обидва підходи забезпечують хорошу оцінку для виявлення відхилень в ЕМГ, що робить їх корисними інструментами в аналізі та діагностиці м'язових порушень і можуть зменшувати помилку класифікації до 10%

```
def getDAR(vector, order=4):
    # Get the first difference of the vector
    vector_diff = np.diff(vector)
    # Calculate the AR coefficient on it
    return getAR(vector_diff, order=4)
```

```
def getAR(vector, order=4):
    # Using Levinson Durbin prediction algorithm, get autoregressive coefficients
    # Square signal
    vector = np.asarray(vector)
    R = [vector.dot(vector)]
    if R[0] == 0:
        return [1] + [0]*(order-2) + [-1]
    else:
        for i in range(1, order + 1):
            r = vector[i:].dot(vector[:-i])
            R.append(r)
        R = np.array(R)
        # step 2:
        AR = np.array([1, -R[1] / R[0]])
        E = R[0] + R[1] * AR[1]
        for k in range(1, order):
            if (E == 0):
                E = 10e-17
            alpha = - AR[:k + 1].dot(R[k + 1:-1]) / E
            AR = np.hstack([AR, 0])
            AR = AR + alpha * AR[:-1]
            E *= (1 - alpha ** 2)
        return AR

return AR_coeff
```

```
def getCC(vector, order =4):
    AR = getAR(vector, order)
    cc = np.zeros(order+1)
    cc[0] = -1*AR[0]# issue with this line
    if order > 2:
        for p in range(2,order+2):
            for l in range(1, p):
                cc[p-1] = cc[p-1]+(AR[p-1] * cc[p-2] * (1-(l/p)))

    return cc
```

8. Характеристики стандартного відхилення похідної сигналу - Середньоквадратичне значення дає загальне значення поведінки сигналу протягом досліджуваного інтервалу часу. Робота [44] показує,

що такі характеристики є надійною ознакою, яка може бути використана для ідентифікації домінуючих м'язових волокон, залучених для жесту. Також, обчислювався міжквартильний розмах – пов'язаний з середніми значеннями сигналу і який є характеристикою скорочення м'язів, бо дозволяє аналізувати викиди сигналу.

```
def getDASDV(vector):  
    vector = np.asarray(vector)  
    return np.lib.scimath.sqrt(np.mean(np.diff(vector)))  
  
def getIQR(vector):  
    vector = np.asarray(vector)  
    vector.sort()  
    return [vector[int(round(vector.shape[0]/4))], vector[int(round(vector.shape[0]*3/4))]]
```

9. Box counting fractal dimension – у [45] показано, що фрактальні характеристики можуть покращувати якість розпізнавання сигналів м'язів, через те, що вони демонструють сліди самоподібності, а також через здатність фрактальних розмірностей характеризувати та вимірювати складність, притаманну різним типам м'язового скорочення. Для реалізації box counting методу, «бокси» різних розмірів накладаються на зображення у фіксованій сітці. Кількість прямокутників, що охоплюють край сигналу (тобто частково 1 і частково 0), підраховується як функція розміру рамки. Цей підрахунок потім використовується для обчислення фрактальної розмірності.

```
def getBC(vector):  
    k_max = int(np.floor(np.log2(len(vector))))-1  
  
    Nr = np.zeros(k_max)  
    r = np.zeros(k_max)  
    for k in range(0, k_max):  
        r[k] = 2**(k+1)  
        curve_box = int(np.floor(len(vector)/r[k]))  
        box_r = np.zeros(curve_box)  
        for i in range(curve_box):  
            max_dat = np.max(vector[int(r[k]*i):int(r[k]*(i+1))])  
            min_dat = np.min(vector[int(r[k]*i):int(r[k]*(i+1))])  
  
            box_r[i] = np.ceil((max_dat-min_dat)/r[k])  
        Nr[k] = np.sum(box_r)  
  
    bc_poly = np.polyfit(np.log2(1/r), np.log2(Nr), deg=1)  
    return bc_poly[0]
```

10. Середнє значення другої похідної функції є мірою середньої швидкості зміни градієнта функції на заданому інтервалі і також пов'язана з інтенсивністю м'язової напруги. Використовувалася у [46] при аналізі перехресних перешкод (electrical cross-talk).

```
def getDASDV(vector):  
    vector = np.asarray(vector)  
    return np.lib.scimath.sqrt(np.mean(np.diff(vector)))
```

Деякі з перелічених особливостей віддають декілька значень, тому довжина вектора особливостей більша, ніж кількість функцій, які застосовувалися. За допомогою знаходження особливостей вдалося зменшити отримані раніше нарізані сигнали до розмірів 22 особливості x 28 каналів, тобто суттєво зменшити кількість часу, який потрібен на навчання. Такий вибір розподілу та знаходження особливостей має свої переваги: протягом навчання нейронна мережа буде аналізувати патерни не всього сигналу загалом, а окремих його частин, враховуючи, що особливості можуть змінюватися протягом 5 секунд запису і тем не менш відповідати одному і тому самому жесту.

3.3 Нейронні мережи

Нейронна мережа (НМ) — це тип штучного інтелекту, в якому частково імітуються дії мозку людини. НМ створює зв'язки між елементами обробки, за допомогою математичного еквівалента реальних нейронів. НМ гарно підходять для прогнозування подій, коли мережі мають великий набір «вхід-вихід» на основі яких можна проводити порівняльне навчання. Це дуже зручний математичний апарат для обробки великої кількості даних та знаходження складних, неявних зв'язків між даними, які неможливо відстежити аналітично. Нейронні мережі — це набір математичних функцій (або алгоритмів) які можуть виявляти патерни подібно до того, як це може робити мозок людини. Цей набір функцій приймає дані у вигляді масивів чисел у які мають бути переведені всі реальні дані, наприклад зображення (попільно), сигнали або навіть текст. У нашому мозку, який, суто кажучи, є набором нейронів, кожен з них приймає електричні та хімічні сигнали як

вихідні дані через численні дендрити та передає вихідні сигнали через свій аксон (насправді, є винятки, як у мультиполярних нейронів). Аксони утворюють контакти з іншими нейронами в з'єднаннях, які називаються синапсами, де вони передають свої вихідні сигнали іншим нейронам, щоб повторювати той самий процес знову, таким чином відтворюючи декілька десятків трильйонів зв'язків у мозку людини. Тож, штучна НМ, подібно до мозку, являє собою набір зв'язаних елементів, які також називають нейронами. З'єднання між нейронами передає сигнали між ними. Нещодавно машинне навчання (МН) стало дуже поширеним у наукових дослідженнях і було впроваджене в різноманітні застосування, включаючи завдання класифікації, видобування текстової інформації, виявлення спаму, рекомендації відео, класифікацію зображень і мультимедійний пошук [47]. Серед різних алгоритмів МН, глибоке навчання (ГН, Deep Learning) є дуже поширеним у цих застосуваннях [48]. Постійне зростання нових досліджень у галузях глибокого та розподіленого навчання пов'язано як і з непередбачуваним зростанням можливості отримання різноманітних даних, так і з вражаючим прогресом у технологіях апаратного забезпечення.

Для розуміння нотації, яка використовується у цій роботі, треба виділити основні складові нейронної мережі. Нейрон – це обчислювальна одиниця (або математична функція), яка отримує числову інформацію, проводить з нею певні маніпуляції та передає далі. Найбазовіше уявлення про нейронну мережу можна скласти, якщо відноситися до неї, як до певної математичної моделі або функції, яка ставить у відповідність набору вхідних даних набір вихідних значень. Якщо ми маємо готовий набір вхідних та вихідних даних, то вихідні значення повинні бути якнайближчими до набору очікуваних значень, тобто функція повинна бути певним чином оптимізована (навчена). Після закінчення процесу оптимізації по відомому набору даних, отримана функція може використовуватися для вирішення задач, аналогічних тим, на яких вона навчалася.

Навчання можна розділити на навчання «з учителем» і «без учителя». Навчання з учителем: як зрозуміло з назви, цей тип навчання здійснюється під наглядом. Цей процес навчання залежить від коригування. Під час тренування нейронної мережі за допомогою навчання з учителем набір вхідних даних подається в мережу, яка генерує вихідні дані. Цей вихідний набір порівнюється з бажаним (тестовим) набором. Якщо є відхилення між фактичним і бажаним виходами, генерується сигнал помилки. На основі цього сигналу помилки, ваги мережі коригуються до тих пір, поки фактичний вихід не стане таким же (або майже таким же), як бажаний.

Навчання без учителя – здійснюється без нагляду. Це – незалежне навчання. Коли подається новий вхідний набір, нейронна мережа дає вихідні дані, які дають певний результат, але немає зворотного зв'язку з бажаним набором про те, який має бути вихід і чи потрібно його корегувати. Тому при такому принципі навчання (який потрібен, коли оператор не має інформації типу «вхід-вихід») мережа сама повинна виявляти патерни та особливості у вхідних даних (що власне і є основним об'єктом дослідження при такому типі навчання).

Найпростіший спосіб знаходження відповідності між вхідними та вихідними даними – це модель лінійної регресії. Нехай X – це набір певних даних та Y_{known} – це відомі нам правильні вихідні дані, W – це матриця вагів, які накладаються на ці дані, тоді

$$X \times W + B = Y,$$

де Y – набір вихідних значень, а B (bias) зміщення. Щоб реалізувати «навчання» треба підібрати найбільш оптимальні значення W та B . Це можна зробити, аналізуючи функцію втрат (Loss function), яку можна обрати, наприклад, у вигляді у середньоквадратичної помилки (mean squared error, MSE).

$$MSE = MSE(Y[W, B], Y_{known})$$

Після обчислення градієнту цієї функції, який фактично буде залежати від

значень W та B , ми повинні цих зробити оновлення даних, наприклад за допомогою метода градієнтного спуску:

$$W_{i_new} = W_i - LR \times \frac{\partial MSE(Y[W, B], Y_{known})}{\partial W_i}$$

$$B_{i_new} = B_i - LR \times \frac{\partial MSE(Y[W, B], Y_{known})}{\partial B_i}$$

Де LR (learning rate) – швидкість навчання, яка може бути постійною або корегуватися у процесі навчання, якщо, наприклад, є ризики «пропустити» мінімум. У такої моделі є очевидний мінус – вона може враховувати тільки лінійний взаємозв'язок між даними, тоді як значна кількість досліджуваних зв'язків є вочевидь суттєво нелінійними. Це обмеження можна зняти, якщо на результати обчислення подіяти нелінійною функцією, яка буде називатися функцією активації, яке визначає вихід кожного елемента (перцептрона або нейрона) в нейронній мережі. Вона приймає «вхід» від кожного нейрона і перетворює його у «вихід», зазвичай у діапазоні від 0 до 1 або від -1 до 1. Її можна визначити як додаткове дію, прикладену до входу, щоб отримати точніший результат. У НМ кожен нейрон має вагу і вхідні дані матрично множаться на вагу та надаються у функцію активації. Вихід кожного нейрона є входом нейронів наступного шару мережі, і тому вхідні дані каскадують через кілька функцій активації, доки врешті вихідний шар не згенерує певний прогноз. Лінійна модель разом з функцією активації має назву перцептрона (рис. 3.4). Тобто перцептрон – це модель нейронної мережі з одним нейроном.

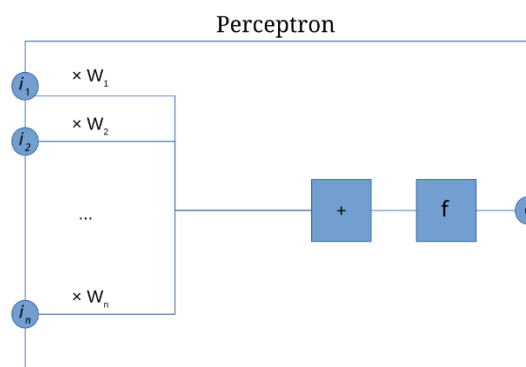


Рисунок 3.4 – Перцептрон. Накладання вагів, зміщення та обчислення функції активації від вхідних даних та повернення вихідних даних

Найбільш поширеними функціями активації є:

1. Сігмоїда (рис. 3.5) - Ця функція приймає будь-яке дійсне число як вхід і повертає значення у діапазоні від 0 до 1. Чим більший (більш позитивний) вхід, тим ближче значення виходу до 1, а чим менший (більш негативний) вхід, тим ближче значення виходу до 0.

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

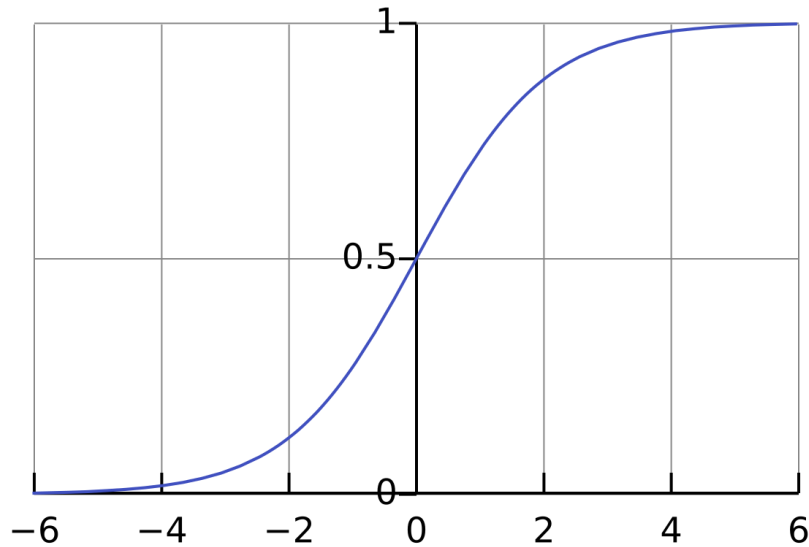


Рисунок 3.5 – Сігмоїда

Вона забезпечує гладку зміну вихідного значення, що є корисним для стабільності навчання моделі. Вона є диференційованою, що необхідно при використанні методу зворотного проходу помилки під час оптимізації. Завдяки своїй S-подібній формі, сигмоїда добре підходить для задач, де потрібно розмежувати активацію між малими та великими вхідними значеннями, і це корисно для задачі класифікації або роботи з нелінійними даними.

2. Гіперболічний тангенс (рис. 3.6) - є варіантом сигмоїда, який змінюється від -1 до +1. Це допомагає зробити вихід більш симетричним відносно нуля, що сприяє покращенню градієнтів для значень, різних за знаком.

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

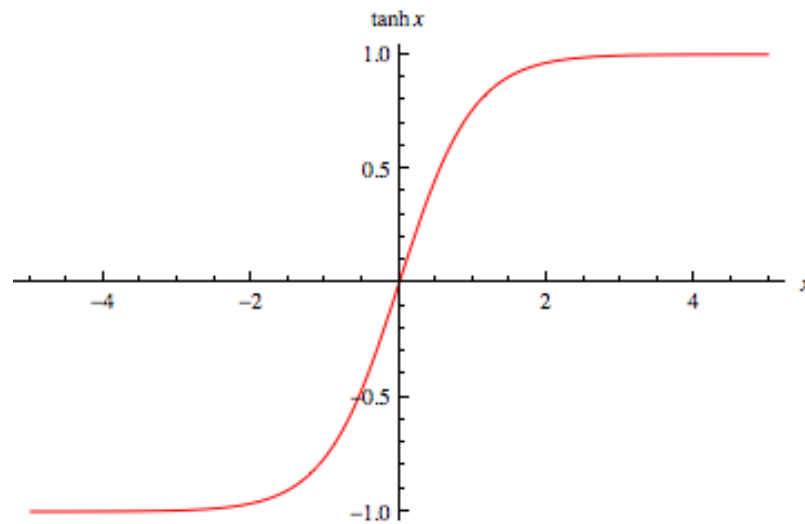


Рисунок 3.6 – Гіперболічний тангес

Гіперболічний тангес забезпечує гладкий перехід між активаціями, а його похідна досягає має максимум поблизу нуля, що ефективніше для оновлення ваг. Завдяки цим властивостям \tanh часто використовується у Deep Learning, особливо в моделях, які потребують більш чутливих активацій для нелінійного представлення даних (як і сігмоїда).

3. *ReLU* (рис. 3.7) (rectified linear unit) – функція, яка просто повертає вхідне значення, якщо воно більше за нуль та повертає нуль, якщо менше:

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

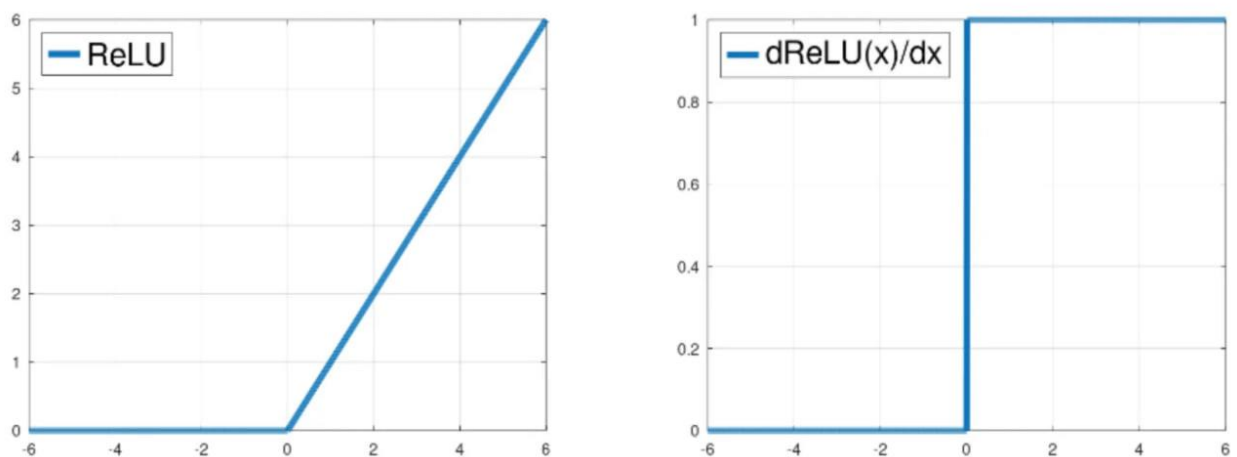


Рисунок 3.7 – ReLU та її похідна

Переваги використання *ReLU* функції активації полягають у її простоті та ефективності обчислень. Завдяки своїй формі *ReLU* сприяє швидшій і

ефективнішої збіжності моделей під час навчання, оскільки забезпечує постійний градієнт для позитивних значень, на відміну від сигмоїдної функції або гіперболічного тангенсу, розглянутих раніше, де градієнт може ставати малим (згасання градієнту), через насичення функції. Крім того, *ReLU* сприяє створенню «розрідження», активуючи лише позитивну частину нейронів, що зменшує взаємозалежність між нейронами та сприяє підвищенню продуктивності моделі. Існують дослідження [49], що демонструють, що для цілей Deep Learning, *ReLU* дає більшу точність аніж сигмоїда та гіперболічний тангенс.

Шар у нейронній мережі — це група нейронів, які об'єднані і виконують певну обробку даних паралельно. Тобто один шар міститься з моделі лінійної регресії та функції активації. Усі нейрони одного шару працюють одночасно і передають свої виходи до наступного шару. Структура нейронної мережі складається з кількох шарів, які працюють разом для обробки та перетворення вхідних даних на вихід (рис. 3.8). Ці шари зазвичай організовані ієрархічно і класифікуються на три основні типи: вхідний шар, приховані та вихідний шари.

Вхідний шар є першим шаром мережі, в який подаються необроблені дані. Кожен нейрон у вхідному шарі представляє одну ознаку вхідних даних і передає їх далі. Кількість нейронів у вхідному шарі прямо залежить від кількості ознак у вхідному наборі даних. Наприклад, у завданні класифікації зображень кожен піксель на зображенні відповідав би одному нейрону на вхідному шарі (хоча такий підхід використовується нечасто), тому в нашій задачі ми зменшували кількість вхідних даних, щоб зменшити кількість вхідних нейронів. Після вхідного і до вихідного шару йдуть приховані шари. Нейронна мережа може мати один або кілька прихованих шарів, і кожен шар зазвичай містить кілька нейронів. Вони потрібні для виконання складних перетворень шляхом застосування ваг, зсувів і різноманітних функцій активації до вхідних даних з попереднього шару. Кількість прихованих шарів і нейронів у кожному з них визначається складністю завдання та архітектурою

мережі. Приховані шари потрібні для вивчення характерних патернів основних зв'язків в даних. Однак збільшення шарів призводить до збільшення часу навчання і може призвести до перенавчання, що буде описано далі.

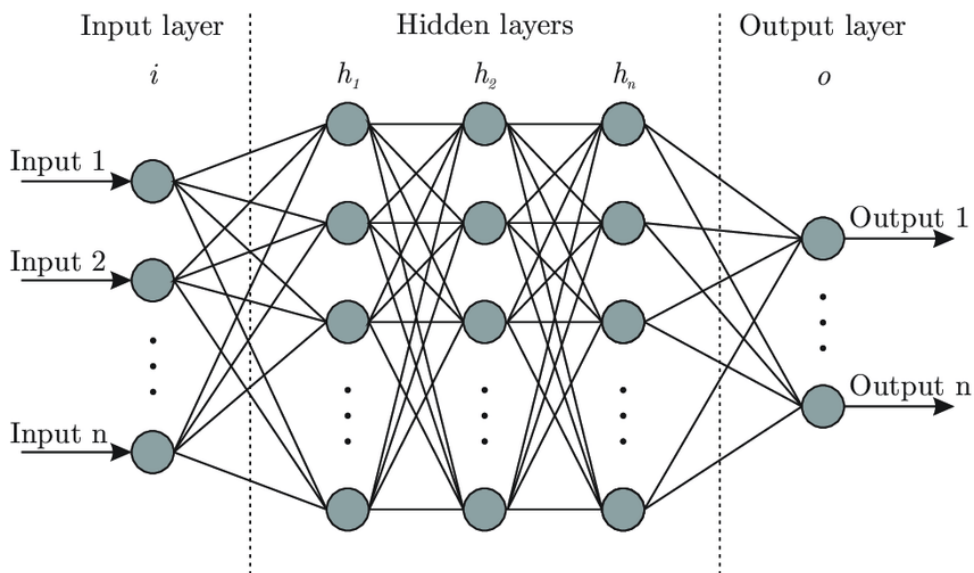


Рисунок 3.8 – Нейронна мережа та її структура – вхідний шар, приховані шари та вихідний шар

Нарешті, вихідний шар генерує кінцевий прогноз (результат класифікації). Вихідний шар має один або кілька нейронів залежно від завдання. У задачах з великою кількістю класів вихідний шар може мати кілька нейронів, кожен з яких відповідає одному класу. У вихідному шарі зазвичай використовується функція активації softmax (для багатокласової класифікації) або sigmoid (для бінарної класифікації), для створення ймовірнісного розподілу по можливих класах. Загальна структура нейронної мережі — від вхідного шару через приховані шари до вихідного — дозволяє мережі обробляти дані таким чином, що це нагадує роботу людського мозку, що дає змогу вчити складні патерни та робити точні прогнози чи класифікації.

Якщо НМ міститься з багатьох шарів, то навчання по такій мережі називають глибоким (Deep Learning). Треба виділити поняття повнозв'язаного шару (Fully Connected Layer) — це тип шару, в якому кожен нейрон цього шару з'єднаний з усіма нейронами попереднього шару. Тобто, кожен нейрон

приймає вхід від кожного нейрона попереднього шару та передає сигнал на кожен нейрон наступного шару. Ці з'єднання між нейронами визначаються вагами, які в процесі навчання ітеративно коригуються.

Нейронна мережа прямого зв'язку (Feedforward neural network)— тип штучної НМ, у якій інформація рухається лише в одному напрямку, від вхідного рівня через будь-які приховані рівні й, нарешті, до вихідного рівня. Термін «прямий зв'язок» означає той факт, що з'єднання між блоками не утворюють цикл, на відміну від більш складних типів мереж.

Прямий прохід (forward propagation or forward) — це процес, за допомогою якого вхідні дані проходять через рівні нейронної мережі для створення виходу. В результаті обчислюються передбачуваний вихід для заданого вхідного сигналу, пропускаючи його через мережу шар за шаром, застосовуючи математичні операції на кожному кроці. Під час прямого поширення мережа обчислює зважену суму вхідних даних для кожного нейрона, застосовує функції активації для введення нелінійності та пропускає виходи через послідовні рівні, доки не буде досягнуто остаточний результат.

Зворотній прохід (Backward propagation or backpropagation or backward)— це ключовий алгоритм, який використовується для навчання нейронних мереж шляхом оновлення вагових коефіцієнтів мережі на основі аналізу похибки між прогнозованим результатом і фактичними цільовими значеннями. Процес починається після прямого проходу мережі, де вхідні дані передаються через мережу для створення прогнозів. Коли прогнози зроблені, вони порівнюються з істинними значеннями, а помилка обчислюється за допомогою функції помилок Loss. Це робиться шляхом застосування ланцюгового правила диференціювання для поширення помилки назад від вихідного рівня до вхідного. Для кожного шару в мережі обчислюється градієнт похибки, який показує, наскільки б вона змінилася б, якщо певну вагу було б скориговано. Для одного шару $z = Wx + b$ з функцією активації $a = \sigma(z)$ градієнт функції L втрат за функцією активації:

$$\delta_L = \frac{\partial L}{\partial a}$$

І градієнт активації:

$$\delta_\sigma = \frac{\partial \sigma}{\partial z} = \sigma'(z)$$

Отже сигнал помилки для шару:

$$\delta = \delta_L \times \delta_\sigma$$

Градієнт ваги можна обчислити використовуючи δ

$$\frac{\partial L}{\partial W} = \delta \times x^T$$

Транспонування виникає по правилам множення матриць, що неодноразово доведено, наприклад, у [50].

Градієнт зміщення:

$$\frac{\partial L}{\partial b} = \sum \delta$$

Для багатошарової мережі цей процес проводиться аналогічно:

Для вихідного шару (індекс L вказує на те, що мова йде про останній шар НМ):

$$\delta^{(L)} = \frac{\partial L}{\partial a^{(L)}} = \sigma'(z^{(L)})$$

Для прихованих шарів (тут слід зазначити, що помилка шару l залежить від помилки шару $l + 1$ через зворотне поширення помилки, яка рухається від вихідного шару мережі до вхідного):

$$\delta^{(l)} = (W^{(l+1)})^T \delta^{(l+1)} \sigma'(z^{(l)})$$

Де $\delta^{(l+1)}$ – помилка наступного ($l + 1$) шару яка вже була обчислена на попередньому кроці, $W^{(l+1)}$ – матриця ваг, що з'єднує поточний шар l із наступним шаром ($l + 1$), $\sigma'(z^{(l)})$ – похідна функції активації для поточного шару l .

Роблячи це, мережа дізнається, який внесок кожного нейрона в кожному шарі в кінцеву помилку. Оновлення ваги здійснюється шляхом, наприклад, віднімання частки обчислених градієнтів від поточних ваг, як це було раніше показано. Ця частка визначається швидкістю навчання, гіперпараметром LR ,

який контролює швидкість процесу навчання. Вагові коефіцієнти оновлюються кожний проход з метою мінімізації помилок з часом. Той самий процес застосовується до зміщень у мережі, які також коригуються на основі градієнтів, обчислених для кожного шару.

Зворотне розповсюдження виконується протягом багатьох ітерацій (epoch), і з кожною ітерацією ваги та зміщення мережі налаштовуються для покращення прогнозів. Цей ітераційний процес поступово зменшує помилку, дозволяючи мережі вивчати оптимальні параметри для створення точних прогнозів на нових, невідомих даних.

Слід відзначити ще два важливих терміни, які згадувалися раніше: Перенавчання (overfitting) – негативний результат роботи з НМ, коли модель настільки налаштована на тренувальні дані, що вона починає вивчати не тільки загальні закономірності, а й випадкові шуми або специфічні особливості цих даних. В результаті, модель може дуже точно передбачати результати для даного набору тренувальних даних, але погано працює на нових, раніше невідомих даних (тестових), це може відбутися через надмірну складність моделі (наприклад, занадто багато параметрів або шарів у нейронній мережі) або недостатню кількість даних для навчання.

Недонавчання (underfitting) – також негативна ситуація, коли модель не змогла знайти та вивчити достатню кількість закономірностей у даних, і її здатність робити точні прогнози є слабкою як для тренувальних, так і для тестових даних. Така модель занадто проста або має недостатньо параметрів, щоб ефективно відображати складні структури в даних.

3.4 Модель Deep Learning для розпізнавання жестів

Таким чином, в нашій роботі використовувалася тришарова повнозв'язна нейронна мережа прямого зв'язку для глибокого навчання з учителем, яка приймає на вхід $22 \times 28 = 616$ значень «особливостей» і повертає ймовірності того, що даний сигнал належить даному класу. На рисунку 3.9 наведена структура використаної нейронної мережі, аналогічна до рисунка 3.8.



Рисунок 3.9 – Графічна структура моделі: 616 вхідних значень, 128 значень у прихованому шарі, 34 у вихідному і 17 значень (номерів жестів) у результаті.

Нижче представлений клас *Net*, який успадковується від *nn.Module* — базового класу для всіх нейронних мереж у бібліотеці *PyTorch*, що означає, що він отримує всі методи та атрибути від базового класу *PyTorch*, такі як автоматичне обчислення градієнтів, збереження та завантаження моделі та інші корисні функції

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(22*28, 128)
        self.dropout = nn.Dropout(0.25)
        self.fc2 = nn.Linear(128, 34)
        self.fc3 = nn.Linear(34, 17)

    def forward(self, x):
        x = torch.flatten(x, 1)
        x = self.fc1(x)
        x = F.relu(x)
        x = self.dropout(x)
        x = self.fc2(x)
        x = F.relu(x)
        x = self.dropout(x)
        x = self.fc3(x)
        output = F.log_softmax(x, dim=1)
        return output

model = Net()
```

Клас у *Python* — це структура, яка описує об'єкт з певними властивостями (атрибутами) та методами. У контексті нейронних мереж клас є шаблоном для створення та опису структури мережі. У нашому прикладі клас *Net* описує архітектуру нейронної мережі, яка буде навчатися для виконання задачі класифікації. Класи використовуються по декільком причинам: всі компоненти нейронної мережі (шари, функції активації, методи обчислень) об'єднані в одному об'єкті, що дозволяє зручно керувати мережею та працювати з нею як з єдиним цілим; використання класів допомагає зробити

код більш організованим та зручним для розробки та тестування, бо вся логіка нейронної мережі зібрана в одному місці. Наведемо опис деяких ділянок програми: `Torch.flatten(x, 1)` — це функція, яка перетворює вхідні дані x в одновимірний вектор. `Self.dropout = nn.Dropout(0.25)` — шар Dropout, який допомагає уникнути перенавчання, випадковим чином з ймовірністю 25% відключаючи нейрони під час навчання. Через зазначені в минулому розділі переваги, у вхідному та прихованому шарах було вирішено використовувати функцію активації *ReLU*. У вихідному шарі вирішено використовувати функцію *log_softmax*. Функція *softmax* (рис. 3.10) діє на кожне значення z_i K -вимірному вектора z експоненційною функцією і нормалізує на суму значень всіх таких експонент:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

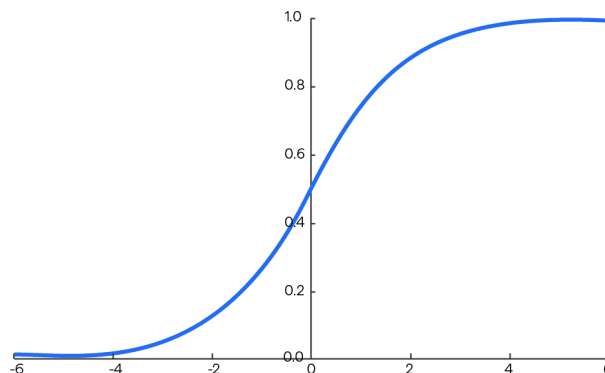


Рисунок 3.10 – Функція softmax

Ця функція перетворює K -вимірний на вектор ймовірностей кожного значення, сильно виділяючи максимальну ймовірність, за що має таку назву. Функція активації *log_softmax* поєднує в собі два етапи: застосування функції *softmax* і взяття логарифма від отриманих значень. У функції *log_softmax* логарифм обчислюється одночасно з функцією *softmax*, що запобігає виникненню проблеми числового переповнення або втрати точності, яка може трапитися при роботі з експонентами у класичній *softmax*. Це особливо важливо для великих або малих значень z , які можуть спричинити помилки під

час обчислень. Логарифм ймовірностей часто зручніший для градієнтного спуску, оскільки логарифми перетворюють добутки на суми, спрощуючи обчислення градієнтів. Тому у задачах з великою кількістю класів (наприклад, класифікація) градієнти обчислюються ефективніше. Завдяки тому, що функція обмежує значення логарифмічних ймовірностей, моделі, які використовують *log_softmax*, мають кращу стабільність під час навчання, особливо для великих наборів даних або моделей із великою кількістю параметрів. Через те, що *Log_softmax* видає логарифм ймовірностей, що добре поєднується з негативною логарифмічною правдоподібністю, яка буде використана при навчанні.

Навчання нейронної мережі *Net* відбувається за допомогою бібліотеки *PyTorch* для класифікації жестів, що включає в себе використання функції втрат *CrossEntropyLoss* та оптимізатора *Adam*:

```
import torch.optim as optim

cross_entropy = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
for epoch in range(3000):
    for l in range(38):
        # Zero gradients
        optimizer.zero_grad()
        # TODO: write a function prepare_data() which loads and prepares the
        # data for participant i.
        # Forward pass
        predictions = model(
            torch.tensor(np.real(np.load(f»Participant_{l +
1}_features.npy»)), dtype=torch.float32, requires_grad=False))
        # Compute loss
        loss = cross_entropy(predictions,
torch.tensor(np.real(np.load(f»Participant_{l + 1}_gestures.npy»)),
dtype=torch.long,
                                                    requires_grad=False))

        # Backpropagation
        loss.backward()
        # Optimizer step
        optimizer.step()

    print(f»Epoch {epoch}, Loss: {loss.item()}»)

torch.save(model.state_dict(), «model_diploma.pt»)
```

В якості функції втрат використовувалася функція крос-ентропії (*Cross-Entropy*). Вона часто використовується в задачах класифікації, завдяки обчисленню різниці між розподілами ймовірностей для всіх класів одночасно. Вона вимірює різницю між двома розподілами ймовірностей: розподілом

передбачень моделі (p) та справжніми мітками (q). Якщо q_i – це справжня ймовірність для класу i (зазвичай це 1 для правильного класу та 0 для інших класів), а p_i – ймовірність, передбачена моделлю для класу i , то функція кросс-ентропії має вигляд:

$$\text{CrossEntropyLoss} = - \sum_{i=1}^c q_i \log(p_i)$$

Завдяки використанню логарифму ймовірностей модель «штрафується» сильніше за неправильні передбачення, які мають високий рівень впевненості. Це дозволяє досягати кращої узгодженості в навчанні. Ця функція втрат формує градієнти таким чином, щоб ваги оновлювались пропорційно до похибки моделі для кожного класу, з урахуванням їхньої ймовірності. Це допомагає швидше досягти оптимальних результатів. Оскільки градієнти функції добре визначені та гладкі, вона забезпечує стабільний процес навчання, без стрибків у значеннях втрат.

У якості оптимізатора (*optimizer*) був обраний *optim.Adam*. Оптимізатор – це алгоритм, який оновлює ваги моделі, щоб мінімізувати функцію втрат. Оптимізатори керують тим, як ваги змінюються під час навчання на основі обчислених градієнтів. *Adam* (Adaptive Moment Estimation) – один з найпопулярніших оптимізаторів у глибокому навчанні, який об'єднує та використовує найкращі сторони двох інших методів: *Momentum* та *RMSProp*. Перший дозволяє враховувати попередні градієнти, щоб прискорити та уточнити навчання, другий ефективно адаптує швидкість навчання залежно від градієнтів для кожного параметра lr , початкове значення якого ми обрали 0.01.

Кроки оновлення ваги у оптимізаторі Adam:

1. Обчислення першої моментної оцінки, для врахування напрямку руху:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

Де, t – ітерація, m_t – експоненційно зважене середнє значення градієнтів (перша моментна оцінка), β_1 (та β_2 далі) коефіцієнти зміщення для моментів

(за замовчуванням $\beta_1 = 0.9$, $\beta_2 = 0.999$), $g_t = \frac{\partial W}{\partial w_t}$ – градієнт функції втрат по вазі w_t .

2. Друга моментна оцінка (середнє квадратів градієнтів), для оцінки масштабу:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Корекція зміщення для моментів:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Вага w_{t+1} для кожної наступної ітерації оновлюється наступним чином:

$$w_{t+1} = w_t - lr_{eff_t} \times \hat{m}_t$$

Причому ефективний крок навчання змінюється при кожній ітерації, як:

$$lr_{eff_t} = \frac{lr}{\sqrt{\hat{v}_t} + \epsilon}$$

Де ϵ мале число (за замовчуванням 10^{-8}).

Завдяки адаптивному масштабу градієнтів *Adam* добре працює в умовах високого рівня шуму або варіативності даних, зберігаючи стабільність процесу навчання, що дуже добре підходить до нашої задачі, бо для збереження інформації для особливостей, було вирішино не фільтрувати сигнал. *Adam* є гарним методом для застосування на практиці, оскільки його гіперпараметри (β_1 , β_2 , ϵ) вже мають стандартні, перевірені значення, які працюють добре для більшості задач навчання. Це знижує потребу у складному попередньому налаштуванні.

Навчання відбувалося на даних 38 учасників, протягом 3000 епох, причому оновлення ваги в рамках кожної епохи відбувалося для кожного учасника завдяки циклу по номеру l учасника, за що відповідає цикл *for l in range(38)*.

4. РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

В результаті навчання, фінальні значення Loss набули наступне значення для 38 учасників (Таблиця 1) (Loss розташовані від першого до 38 учасника):

Таблиця 4.1 – Loss в останню епоху навчання

Epoch 2999, Loss: 1.036230206489563	Epoch 2999, Loss: 0.9443536996841431
Epoch 2999, Loss: 1.0278196334838867	Epoch 2999, Loss: 1.0377171039581299
Epoch 2999, Loss: 1.267091155052185	Epoch 2999, Loss: 0.7184184789657593
Epoch 2999, Loss: 1.2220126390457153	Epoch 2999, Loss: 0.5469343066215515
Epoch 2999, Loss: 0.9062356352806091	Epoch 2999, Loss: 0.7414090037345886
Epoch 2999, Loss: 1.2072261571884155	Epoch 2999, Loss: 0.724238395690918
Epoch 2999, Loss: 0.6647678017616272	Epoch 2999, Loss: 0.9236742258071899
Epoch 2999, Loss: 0.9857528209686279	Epoch 2999, Loss: 0.6934235095977783
Epoch 2999, Loss: 0.8881827592849731	Epoch 2999, Loss: 0.4709862172603607
Epoch 2999, Loss: 0.7350736856460571	Epoch 2999, Loss: 0.8613368272781372
Epoch 2999, Loss: 0.8392530679702759	Epoch 2999, Loss: 0.4622909128665924
Epoch 2999, Loss: 0.8211753964424133	Epoch 2999, Loss: 0.7633150815963745
Epoch 2999, Loss: 1.0160667896270752	Epoch 2999, Loss: 0.9815592765808105
Epoch 2999, Loss: 1.1161116361618042	Epoch 2999, Loss: 0.6047483682632446
Epoch 2999, Loss: 1.1401326656341553	Epoch 2999, Loss: 0.5781924724578857
Epoch 2999, Loss: 0.7025477290153503	Epoch 2999, Loss: 0.584373414516449
Epoch 2999, Loss: 0.752141535282135	Epoch 2999, Loss: 0.4939442276954651
Epoch 2999, Loss: 0.684029221534729	Epoch 2999, Loss: 0.7711870670318604
Epoch 2999, Loss: 0.6292411088943481	Epoch 2999, Loss: 0.6800379157066345

Як результат роботи НМ, на виході для кожного з 43 учасників отримується набір з 11424 17-вимірних векторів зі значеннями логарфмів вирогідності того, що ділянка сигналу відповідає даному жесту. Найбільше значення вектора відповідає найбільш ймовірному номеру жеста для цього сигналу. Приклад результату роботи НМ:

*tensor([-4.9081, -0.5907, -3.4754, -1.8840, -4.5764, -6.8934, -2.9848, -4.8140, -4.1729, -5.5630, -7.1606, -5.7489, -4.4947, -3.3904, -2.2066, -4.9175, -8.1244],
grad_fn=<SelectBackward0>)*

Далі, застосовуючи легке порівняння,

```

good = 0
for i in range(11424):
    prediction = torch.argmax(model(features)[i])
    if prediction == gestures[i]:
        good +=1
        print(good)
        print(i)
print(good)

```

можна знайти точність оцінки для кожного учасника, дані який використовувалися (Таблиця 2). Результати, отримані для кожного учасника в цілому є пропорційними до значень Loss. Різниця між якістю оцінювання даних вказує на те, що в подібних задачах існує залежність від індивідуальної специфіки м'язових скорочень кожного учасника, від потовиділення, інтенсивності скорочення, якості накладання електродів, відсотку підшкірного жиру. Як учасників з особливостями можна виділити учасників 3, 4, 6 для яких був отриманий найбільший Loss і найменша точність. Середнє значення правильності оцінки становить 80%. Якщо навчання проводити лише по одному учаснику, то точність відповідності НМ, яка враховує особливості м'язових скорочень конкретної людини становить 95-99% в залежності від учасника.

Таблиця 4.2 – Точність класифікації на знайомих даних

Учасник 1	73%	Учасник 20	73%
Учасник 2	72%	Учасник 21	72%
Учасник 3	67%	Учасник 22	75%
Учасник 4	68%	Учасник 23	88%
Учасник 5	75%	Учасник 24	74%
Учасник 6	70%	Учасник 25	73%
Учасник 7	87%	Учасник 26	75%
Учасник 8	74%	Учасник 27	76%
Учасник 9	73%	Учасник 28	93%
Учасник 10	87%	Учасник 29	75%
Учасник 11	70%	Учасник 30	92%
Учасник 12	82%	Учасник 31	74%
Учасник 13	84%	Учасник 32	76%
Учасник 14	78%	Учасник 33	87%
Учасник 15	79%	Учасник 34	88%
Учасник 16	75%	Учасник 35	88%
Учасник 17	73%	Учасник 36	90%
Учасник 18	86%	Учасник 37	88%
Учасник 19	88%	Учасник 38	90%

Звичайно, більш цікавим є вивчення того, як модель себе поводить при аналізі даних, з якими вона незнайома. Вибірка для тестування в нашому

випадку міститься з 5 учасників (з 39 по 43), яких ми не використовували для навчання. Перевірка роботи мережі на цих даних, демонструє наступні результати (Таблиця 3):

Таблиця 4.3 – Точність моделі на тестових даних

Учасник 39	80%
Учасник 40	73%
Учасник 41	78%
Учасник 42	72%
Учасник 43	74%

Середня точність на незнайомій вибірці становить 75% відсотків, що є чудовим результатом. Відповідно до [51], точність вже близько 69% можна вважати задовільною. Таку високу точність можна пов'язати з обраним процесом навчання – нейронна мережа у результаті роботи з даними враховувала, що одному й тому ж самому жесту можуть відповідати різні особливості не тільки від різних учасників, а й зміна особливостей з часом у сигналі, бо навчання відбувалася по сегментам сигналу. Отримані данні свідчать про те, що навіть проста модель нейронної мережі може бути застосована як класифікатор для програми протезу. Метод обробки даних дозволяє мінімізувати час, необхідний для опрацювання сигналу від м'язів – з одного боку, ми суттєво (в 15 разів) зменшили кількість вагів за допомогою аналізу сигналу на «особливості», з іншого боку навчання НМ та її застосування на короткий (150 мс) проміжках сигналу дозволяє швидко (за декілька десятих секунди) реалізувати необхідну дію і кожні 150 мс перевіряти сигнал на зміну, щоб швидко перейти до виконання іншого жесту. НМ має прекрасну точність при навчанні для конкретної людини, але також має можливість розрізняти жести будь-якої людини в результаті свого навчання. Це може бути використано, як базова програма протезу, яка потім може бути швидко донавчена для потреб конкретної людини, що може суттєво зменшити кількість часу і грошей, необхідних на роботу з пацієнтом. Отримана нейронна мережа розрізняє 17 жестів, що набагато більше ніж 5-10 жестів, які в середньому досліджуються [52] з точністю 40-98%. Кількість жестів знижує

класифікаційну спроможність отриманої мережі. На отриманий результат також впливають низькі точності для деяких учасників, що можна проаналізувати за набором їхніх особливостей. Цікавим є питання дослідження впливу окремих «особливостей» на точність, розширення і перевірка нових «особливостей», а також перевірка вкладу сигналів окремих каналів для різних жестів, що буде зроблено в майбутньому. Також, досліджуватися може вплив вибору функцій активації, кількості шарів, методів втрат і оптимізації. З проведеного аналізу літератури можна зробити висновок щодо доцільності використання згорточної мережі, що також буде досліджено у майбутньому.

5. ВИСНОВКИ

В результаті проведеної роботи була опрацьована значна кількість літератури пов'язаної з протезуванням верхніх кінцівок, наявної на даний момент. У вступному розділі наведений огляд існуючих технологій протезування, їхня класифікація за висотою ампутації, типом енергоживлення та принципом роботи, типом зчитування біосигналів людини тощо. Був проведений аналіз проблем галузі, зокрема побудови кінцівок з великою кількістю ступенів свободи, проблеми точності сигналів ЕМГ та ЕЕГ та їхньої залежності від зовнішніх факторів, новітніх способів зворотнього зв'язку протез-людина.

Окремо описано питання класифікації сигналів електроміограми м'язів руки відповідно до жестів. Наведений огляд існуючих рішень, проаналізовані переваги та недоліки підходів різних авторів. Був розроблений метод класифікації жестів на основі аналізу сигналів ЕМГ 43 учасників. Метод ґрунтується на попередньому аналізі сигналу, а саме виявлявлянням його маркерних особливостей за допомогою ручної екстракції так званих «особливостей» сигналу, які пов'язані з м'язовою активністю руки людини. Був опрацьований підхід Машинного Навчання, який був обраний для вирішення задачі класифікації, наведені базові принципи побудови архітектури Deep Learning мереж. Для опрацювання особливостей була створена нейронна мережа глибокого навчання з трьома шарами, з функціями активації *ReLU* та *log_soft.max*, *cross-entropy* функцією втрат та оптимайзером *Adam*. Точність отриманих результатів оцінюється, як задовільна і становить 75% правильно класифікованих жестів для незнайомих даних. Розроблена нейронна мережа може використовуватися для навчання на сигналах однієї людини з точністю 95% та як базова, при швидкому донавчання для кожної конкретної людини в масштабах промислового виробництва протезів, що підвищить їхню доступність для людей з обмеженнями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. World Health Organization. Disability. 2023. Available online: <https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health> (accessed on 25 February 2023).
2. Kay SPJ, Leonard DA. Hand transplantation: Can we balance the risks and benefits? *J Hand Surg (European Volume)* 2023;48(3):208–13.
3. Zuo KJ, Olson JL. The evolution of functional hand replacement: From iron prostheses to hand transplantation. *Plast Surg (Oakv)*. 2014 Spring;22(1):44-51. PMID: 25152647; PMCID: PMC4128433.
4. Lehneis HR, Dickey R. Special considerations: Fitting and training the bilateral upper-limb amputee. In: Bowker JH, Michael JW, editors. *Atlas of limb prosthetics: Surgical, prosthetic, and rehabilitation principles*. 2nd ed. St. Louis, MO: Mosby-Year Book, Inc.; 1992. pp. 311–324
5. National Academies of Sciences, Engineering, and Medicine; Health and Medicine Division; Board on Health Care Services; Committee on the Use of Selected Assistive Products and Technologies in Eliminating or Reducing the Effects of Impairments; Flaubert JL, Spicer CM, Jette AM, editors. *The Promise of Assistive Technology to Enhance Activity and Work Participation*. Washington (DC): National Academies Press (US); 2017 May 9. 4, Upper-Extremity Protheses. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK453290/>
6. Cordella, Francesca & Ciancio, Anna & Sacchetti, Rinaldo & Davalli, Angelo & Cutti, Andrea & Guglielmelli, Eugenio & Zollo, Loredana. (2016). Literature Review on Needs of Upper Limb Prosthesis Users. *Frontiers in Neuroscience*. 10. 10.3389/fnins.2016.00209.
7. Maat B, Smit G, Plettenburg D, Breedveld P. Passive prosthetic hands and tools: A literature review. *Prosthet Orthot Int*. 2018 Feb;42(1):66-74. doi: 10.1177/0309364617691622. Epub 2017 Mar 1. PMID: 28190380; PMCID: PMC5810914.

8. Imbinto I, Peccia C, Controzzi M, et al. Treatment of the partial hand amputation: an engineering perspective. *IEEE Rev Biomed Eng.* 2016;9:32–48.
9. Gemmell KD, Leddy MT, Belter JT, et al. Investigation of a passive capstan based grasp enhancement feature in a voluntary-closing prosthetic terminal device. *Conf Proc IEEE Eng Med Biol Soc.* 2016;2016:5019–5025.
10. Engdahl, S.M., Gonzalez, M.A., Lee, C. *et al.* Perspectives on the comparative benefits of body-powered and myoelectric upper limb prostheses. *J NeuroEngineering Rehabil* 21, 138 (2024).
<https://doi.org/10.1186/s12984-024-01436-4>
11. Carey SL, Lura DJ, Highsmith MJ; CP; FAAOP. Differences in myoelectric and body-powered upper-limb prostheses: Systematic literature review. *J Rehabil Res Dev.* 2015;52(3):247-62. doi: 10.1682/JRRD.2014.08.0192. PMID: 26230500.
12. Igual, C.; Pardo, L.A., Jr.; Hahne, J.M.; Igual, J. Myoelectric Control for Upper Limb Prostheses. *Electronics* 2019, 8, 1244.
<https://doi.org/10.3390/electronics8111244>
13. Sekine M, Tsuchiya N, Kita K, Yu W. Designing and testing a hybrid lightweight shoulder prosthesis. *Annu Int Conf IEEE Eng Med Biol Soc.* 2014;2014:2500-3. doi: 10.1109/EMBC.2014.6944130. PMID: 25570498.
14. Davidson, Judith. (2002). A Survey of the Satisfaction of Upper Limb Amputees with Their Prostheses, Their Lifestyles, and Their Abilities. *Journal of hand therapy : official journal of the American Society of Hand Therapists.* 15. 62-70. 10.1016/S0894-1130(02)50011-9.
15. Saradjian A, Thompson AR, Datta D. The experience of men using an upper limb prosthesis following amputation: positive coping and minimizing feeling different. *Disabil Rehabil.* 2008;30(11):871-83. doi: 10.1080/09638280701427386. PMID: 17852212.
16. Aljuhani WS, Alhussaini AA, Alangari HS, Aljabr SZ, Al Karni AF, Alhumaidan MI. Community perception and acceptance of patients with

- prosthetic devices after disabilities in Saudi Arabia. *J Musculoskelet Surg Res* 2023;7:29-33
17. Marinelli, A. & Boccardo, Nicolò & Tessari, Federico & Di Domenico, Dario & Caserta, Giulia & Canepa, Michele & Gini, Giuseppina & Barresi, Giacinto & Laffranchi, Matteo & De Michieli, Lorenzo & Semprini, Marianna. (2022). Active upper limb prostheses: A review on current state and upcoming breakthroughs. *Progress in Biomedical Engineering*. 5. 10.1088/2516-1091/acac57.
 18. Montagnani F, Controzzi M, Cipriani C. 2015.. Is it finger or wrist dexterity that is missing in current hand prostheses?. *IEEE Trans. Neural Syst. Rehabil. Eng.* 23:(4):600–9
 19. Narumi, S.; Huang, X.; Lee, J.; Kambara, H.; Kang, Y.; Shin, D. A Design of Biomimetic Prosthetic Hand. *Actuators* 2022, 11, 167. <https://doi.org/10.3390/act11060167>
 20. J. H. De Groot, S. M. Angulo, C. G. M. Meskers, H. C. M. Van Der Heijden-Maessen, and J. H. Arendzen, "Reduced elbow mobility affects the flexion or extension domain in activities of daily living," *Clin. Biomech.*, vol. 26, no. 7, pp. 713–717, 2011.
 21. Kyranou I, Vijayakumar S and Erden M S 2018 Causes of performance degradation in non-invasive electromyographic pattern recognition in upper limb prostheses *Front. Neurorobot.*
 22. Marinelli A et al 2020 Performance evaluation of pattern recognition algorithms for upper limb prosthetic applications 8th IEEE RAS/EMBS Int. Conf. for Biomedical Robotics and Biomechatronics (Biorob) (IEEE)
 23. S. Sudarsan, E. Chandra Sekaran, Design and Development of EMG Controlled Prosthetics Limb, *Procedia Engineering*, Volume 38, 2012, Pages 3547-3551, ISSN 1877-7058, <https://doi.org/10.1016/j.proeng.2012.06.409>.
 24. T. S. Poo and K. Sundaraj, "Design and development of a low cost EMG signal acquisition system using surface EMG electrode," *2010 IEEE Asia*

- Pacific Conference on Circuits and Systems*, Kuala Lumpur, Malaysia, 2010, pp. 24-27, doi: 10.1109/APCCAS.2010.5774896.
25. Maibam PC, Pei D, Olikkal P, Vinjamuri RK, Kakoty NM. Enhancing prosthetic hand control: A synergistic multi-channel electroencephalogram. *Wearable Technologies*. 2024;5:e18. doi:10.1017/wtc.2024.13
26. Abdulghani, Ali & Sattar, Muhammed & Al-Sahib, N. (2019). CONTROL ON UPPER LIMB PROSTHETIC HAND BY ELECTROENCEPHALOGRAM (EEG).
27. Roche AD, Bailey ZK, Gonzalez M, Vu PP, Chestek CA, Gates DH, Kemp SWP, Cederna PS, Ortiz-Catalan M, Aszmann OC. Upper limb prostheses: bridging the sensory gap. *J Hand Surg Eur Vol*. 2023 Mar;48(3):182-190. doi: 10.1177/17531934221131756. Epub 2023 Jan 17. PMID: 36649123; PMCID: PMC9996795.
28. Ortiz-Catalan M, Mastinu E, Greenspon CM, Bensmaia SJ. Chronic use of a sensitized bionic hand does not remap the sense of touch. *Cell Rep*. 2020, 33: 108539
29. Pawel Trajdos, Marek Kurzynski, Application of context-dependent interpretation of biosignals recognition to control a bionic multifunctional hand prosthesis, *Biocybernetics and Biomedical Engineering*, Volume 44, Issue 1, 2024, Pages 161-182, ISSN 0208-5216, <https://doi.org/10.1016/j.bbe.2024.01.001>.
30. A. G. Jaramillo and M. E. Benalcázar, "Real-time hand gesture recognition with EMG using machine learning," *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Salinas, Ecuador, 2017, pp. 1-5, doi: 10.1109/ETCM.2017.8247487
31. M. A. Ozdemir, D. H. Kisa, O. Guren, A. Onan and A. Akan, "EMG based Hand Gesture Recognition using Deep Learning," *2020 Medical Technologies Congress (TIPTEKNO)*, Antalya, Turkey, 2020, pp. 1-4, doi: 10.1109/TIPTEKNO50054.2020.9299264.

32. A. G. Jaramillo and M. E. Benalcázar, "Real-time hand gesture recognition with EMG using machine learning," *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Salinas, Ecuador, 2017, pp. 1-5, doi: 10.1109/ETCM.2017.8247487.
33. Jiang, N., Pradhan, A., & He, J. (2024). Gesture Recognition and Biometrics ElectroMyogram (GRABMyo) (version 1.1.0). PhysioNet. <https://doi.org/10.13026/89dm-f662>.
34. T. R. Farrell and R. F. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 15, no. 1, pp. 111–118, 2007.
35. B. Peerdeman et al., "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *Journal of Rehabilitation Research & Development*, vol. 48, no. 6, p. 719, 2011.
36. Côté Allard, Ulysse & Campbell, Evan & Phinyomark, Angkoon & Laviolette, Francois & Gosselin, Benoit & Scheme, Erik. (2020). Interpreting Deep Learning Features for Myoelectric Control: A Comparison With Handcrafted Features. *Frontiers in Bioengineering and Biotechnology*. 8. 10.3389/fbioe.2020.00158.
37. Mohammadreza Asghari Oskoei, Huosheng Hu, Myoelectric control systems—A survey, *Biomedical Signal Processing and Control*, Volume 2, Issue 4, 2007, Pages 275-294, ISSN 1746-8094, <https://doi.org/10.1016/j.bspc.2007.07.009>.
38. Montazerin, M., Rahimian, E., Naderkhani, F. *et al.* Transformer-based hand gesture recognition from instantaneous to fused neural decomposition of high-density EMG signals. *Sci Rep* **13**, 11000 (2023). <https://doi.org/10.1038/s41598-023-36490-w>
39. Josephs, David & Drake, Carson & Heroy, Andrew & Santerre, John. (2020). sEMG Gesture Recognition with a Simple Model of Attention. 10.48550/arXiv.2006.03645.

40. Phinyomark, Angkoon & Thongpanja, S. & Hu, Huosheng & Phukpattaranont, P. & Limsakul, Chusak. (2012). The Usefulness of Mean and Median Frequencies in Electromyography Analysis. 10.5772/50639.
41. Fajardo, Jose & Gomez, Orlando & Prieto, Flavio. (2020). EMG hand gesture classification using handcrafted and deep features. Biomedical Signal Processing and Control. 63. 10.1016/j.bspc.2020.102210.
42. Kang, Wen-Juh & Shiu, Jiue-Rou & Cheng, Cheng-Kung & Lai, Jin-Shin & Tsao, Hen-Wai & Kuo, Te-Son. (1995). The application of cepstral coefficients and maximum likelihood method in EMG pattern recognition. IEEE transactions on bio-medical engineering. 42. 777-85. 10.1109/10.398638.
43. Shokrollahi, M. & Krishnan, Sridhar & Jewell, D. & Murray, Brian. (2010). Autoregressive and Cepstral Analysis of Electromyogram in Rapid Movement Sleep. 10.1007/978-3-642-03882-2_419.
44. K. Veer, T. Sharma, A novel feature extraction for robust EMG pattern recognition, J. Med. Eng. Technol. 40 (4) (2016) 149–154, <http://dx.doi.org/10.3109/03091902.2016.1153739>, arXiv:0912.3973
45. André L.V. Coelho, Clodoaldo A.M. Lima, Assessing fractal dimension methods as feature extractors for EMG signal classification, Engineering Applications of Artificial Intelligence, Volume 36, 2014, Pages 81-98, ISSN 0952-1976, <https://doi.org/10.1016/j.engappai.2014.07.009>.
46. Kilner, J & Baker, Stuart & Lemon, Roger. (2002). A novel algorithm to remove electrical cross-talk between surface EMG recordings and its application to the measurement of short-term synchronisation. The Journal of physiology. 538. 919-30. 10.1113/jphysiol.2001.012950.
47. Rozenwald MB, Galitsyna AA, Sapunov GV, Khrameeva EE, Gelfand MS. A machine learning framework for the prediction of chromatin folding in Drosophila using epigenetic features. PeerJ Comput Sci. 2020;6:307

48. Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP, Shyu ML, Chen SC, Iyengar S. A survey on deep learning: algorithms, techniques, and applications. *ACM Comput Surv (CSUR)*. 2018;51(5):1–36
49. Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60, 84 - 90.
50. Howard B. Demuth, Mark H. Beale, Orlando De Jess, and Martin T. Hagan. 2014. *Neural Network Design* (2nd. ed.). Martin Hagan, Stillwater, OK, USA
51. U. Côté-Allard *et al.*, "Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 4, pp. 760-771, April 2019, doi: 10.1109/TNSRE.2019.2896269.
52. Rajapriya R., Rajeswari K., Thiruvengadam S.J., Deep learning and machine learning techniques to improve hand movement classification in myoelectric control system, *Biocybernetics and Biomedical Engineering*, Volume 41, Issue 2, 2021, Pages 554-571, ISSN 0208-5216, <https://doi.org/10.1016/j.bbe.2021.03.006>.