УДК 004.31:004.8 **DOI:10.30837/rt.2026.1.224.08**

Д.О. ПОГУЛЯЙ, Д.Ю. ГОЛУБНИЧИЙ, канд. техн. наук, М.В. ЄСІНА, канд. техн. наук

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ МОДЕЛЕЙ TINYML ДЛЯ ЗАДАЧ КЛАСИФІКАЦІЇ СЕНСОРНИХ ДАНИХ НА ПЛАТФОРМІ ESP32

Вступ

Традиційно аналіз безперервних потоків сенсорних даних, зокрема просторової інформації з гіроскопів та акселерометрів, вимагав перенесення обчислень на зовнішні хмарні сервери. Хоча штучний інтелект (ШІ) ідеально підходить для пошуку закономірностей у таких великих масивах інформації, класичні нейромережі потребують колосальних обчислювальних потужностей. Автономні пристрої на базі мікроконтролерів, таких як ESP32, не володіють такими ресурсами. Більше того, вони не можуть виділяти всю доступну пам'ять виключно під розпізнавання рухів, оскільки мусять паралельно виконувати інші фонові задачі: опитування периферії, керування моторами чи підтримку мережевого зв’язку. Вирішенням цього конфлікту стала технологія TinyML (Edge AI). Вона дозволяє кардинально стискати моделі ШІ та запускати їх локально, завдяки чому швидко стала стандартом номер один для задач класифікації сенсорних даних. Однак ефективність TinyML напряму залежить від обраної внутрішньої структури нейромережі. Щоб знайти ідеальний баланс між високою точністю та економією ресурсів, розробники змушені порівнювати різні архітектури машинного навчання: від базових повнозв’язних шарів (Dense) до складніших згорткових топологій (1D CNN). Тому детальний порівняльний аналіз цих підходів є критично важливим кроком для створення стабільних, швидких та ресурсоефективних вбудованих систем.

1. Огляд технології Edge Impulse

Платформа Edge Impulse представляє собою комплексне середовище для розробки та розгортання систем TinyML. Її головна перевага полягає в абстрагуванні складних процесів оптимізації базової бібліотеки TensorFlow Lite for Microcontrollers (TFLite Micro). До таких низькорівневих процесів належать: ручний статичний розподіл оперативної пам’яті (управління Tensor Arena), видалення невикористовуваних математичних операторів для зменшення розміру прошивки у Flash-пам’яті (Selective Linking), а також складна конвертація тензорів під цілочисельну арифметику. Завдяки автоматизації цих кроків, інженери отримують змогу зосередитися безпосередньо на архітектурі моделі та якості даних. Типовий пайплайн класифікації сенсорних даних на цій платформі є суворо послідовним і складається з чотирьох ключових етапів: сегментації даних, цифрової обробки сигналів (DSP), машинного навчання та апаратної компіляції.

Сирі дані з сенсора, зокрема з 6-осьового просторового датчика (акселерометр та гіроскоп), надходять у вигляді безперервного потоку. Для того щоб нейромережа могла їх аналізувати, потік розбивається на дискретні блоки фіксованого розміру за допомогою методу ковзного вікна (Sliding Window). Цей метод дозволяє захопити просторово-часовий контекст руху, зберігаючи при цьому фіксовану розмірність вхідного тензора для нейромережі [1].

Подача сирих часових рядів безпосередньо на вхід нейромережі є вкрай неефективною для мікроконтролерів через високу обчислювальну складність. Тому Edge Impulse застосовує блок DSP (Digital Signal Processing) для вилучення ознак (Feature Extraction). Основу цього блоку становить спектральний аналіз, який трансформує сигнал із часової області в частотну за допомогою дискретного перетворення Фур’є (ДПФ):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \quad (1)$$

де $x[n]$ – дискретний вхідний сигнал із сенсора, N – кількість вибірок у вікні, а $X[k]$ – комплексний спектр сигналу [2]. Після обчислення амплітуд та потужностей частот, алгоритм формує компактний вектор спектральних ознак. Це дозволяє кардинально зменшити розмірність вхідних даних без втрати ключової інформації про характер руху.

Отриманий вектор спектральних ознак передається на вхід нейромережі. У базовому варіанті класифікації платформа використовує багатошаровий перцептрон, що складається з повнозв'язних шарів (Dense layers). Математична модель прямого поширення (Forward Propagation) для кожного нейрона в такому шарі описується рівнянням

$$y = f(Wx + b)\#, \tag{2}$$

де x – вектор вхідних ознак (або вихід попереднього шару), W – матриця вагових коефіцієнтів, b – вектор зсуву (bias), а f – нелінійна функція активації (найчастіше ReLU, $f(z) = \max(0, z)$) [3]. На останньому шарі застосовується функція Softmax для перетворення вихідних значень у ймовірності приналежності до кожного з класів жестів.

Головним викликом для мікроконтролера ESP32 є обмежений обсяг оперативної пам'яті (RAM). Для вирішення цієї проблеми Edge Impulse використовує пропрієтарний компілятор EON (Edge Optimized Neural), який компілює нейромережу безпосередньо в оптимізований вихідний код мовою C++, відмовляючись від порівняно важкого інтерпретатора. Додатково застосовується алгоритм квантування після навчання (Post-Training Quantization), який переводить вагові коефіцієнти W з формату з плаваючою комою (Float32) у цілочисельний 8-бітний формат (Int8) за допомогою лінійного відображення:

$$q = round\left(\frac{r}{S}\right) + Z\ \#, \tag{3}$$

де r – реальне значення Float32, q – квантоване значення Int8, S – масштабний множник (Scale), а Z – нульова точка (Zero-point) [4]. Як буде показано у практичній частині дослідження, цей процес може мати нелінійний вплив на фінальну точність моделі (Accuracy) при аналізі сенсорних даних.

2. Топології нейронних мереж для класифікації просторових даних

Ефективність платформи Edge Impulse при розгортанні моделей на мікроконтролерах класу ESP32 напряму залежить від правильно обраної архітектури (топології) штучної нейронної мережі. На відміну від класичних алгоритмів машинного навчання, які часто пропонуються як готові рішення, кастомні нейромережеві архітектури надають розробнику найвищий рівень гнучкості. Оскільки ідеального та універсального рішення для аналізу будь-яких сенсорних даних не існує, інженер завжди змушений шукати оптимальний баланс. У даному дослідженні проводиться глибокий порівняльний аналіз чотирьох принципово різних архітектур. Це дозволить об'єктивно оцінити компроміс між кінцевою точністю розпізнавання просторових рухів (Accuracy) та критичними апаратними витратами, зокрема споживанням оперативної пам'яті (RAM), обсягом згенерованої прошивки (Flash) та часом виконання одного циклу розпізнавання (Inference Time).

Базова повнозв'язна мережа (Shallow Dense) виступає найпростішою формою нейромережевого класифікатора і в межах даної роботи слугує так званою базовою точкою відліку (Baseline) для подальшого порівняння зі складнішими топологіями. Ця модель використовує виключно класичні повнозв'язні шари (Dense layers), у яких кожен штучний нейрон поточного шару має фізичний математичний зв'язок з усіма без винятку нейронами попереднього та наступного шарів. Структурно така мережа формується з трьох ключових етапів. Першим є вхідний шар (Input Layer), розмірність якого суворо відповідає кількості спектральних ознак, попередньо згенерованих на етапі цифрової обробки сигналів (DSP). Далі дані передаються до прихованого шару (Hidden Layer), головним завданням якого є пошук лінійних та нелінійних закономірностей між отриманими частотними характеристиками. Завершується топологія вихідним шаром (Output Layer), який за допомогою функції активації перетворює от-

римані вагові коефіцієнти у відсоткові ймовірності для фінальної класифікації визначених жестів, як наочно показано на рис. 1.

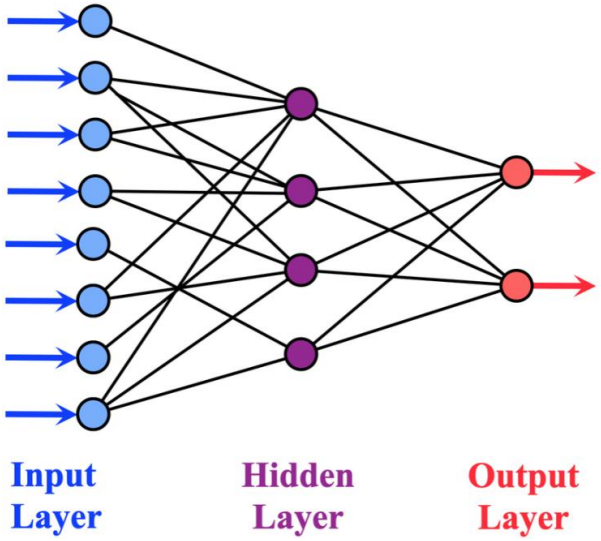

Рис. 1. Загальна топологія базової повнозв'язної нейронної мережі

Головною та беззаперечною перевагою такої базової повнозв'язної моделі є її абсолютна невибагливість до обчислювальних ресурсів. Оскільки процес прямого поширення зводиться до виконання відносно простих операцій матричного множення, оптимізованих компілятором, така мережа здатна приймати рішення за лічені мілісекунди. Крім того, невелика кількість прихованих нейронів гарантує наднизьке споживання дефіцитної оперативної пам'яті мікроконтролера ESP32, залишаючи достатньо вільних ресурсів для паралельного виконання інших важливих системних задач.

Глибока повнозв'язна мережа з регуляризацією (Deep Dense + Dropout) за рахунок пропорційного збільшення ємності (розширення прихованих шарів до 32 та 16 нейронів відповідно) суттєво покращує здатність моделі виявляти складні нелінійні закономірності у спектральних ознаках. Однак при роботі з обмеженими масивами сенсорних даних це створює високий ризик перенавчання (overfitting). За такого сценарію алгоритм жорстко «завчує» тренувальний набір напам'ять, що призводить до різкого падіння точності при спробі класифікувати нові, раніше не бачені рухи. Для нівелювання цієї проблеми та запобігання надмірній коадаптації сусідніх вузлів, до архітектури інтегрується шар просторового відсіву (Dropout). Принцип його математичної дії та безпосередній вплив на кількість активних зв'язків у процесі навчання наочно продемонстровано на рис. 2.

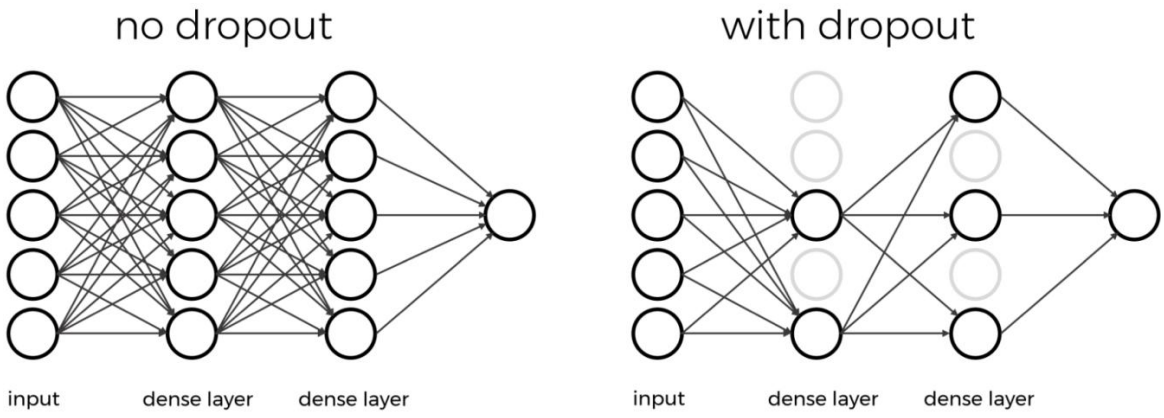

Рис. 2. Візуалізація топології нейронної мережі до та після застосування шару регуляризації Dropout

Математично він працює як множення виходу попереднього шару на бінарний вектор-маску r, елементи якого розподілені за законом Бернуллі з імовірністю p:

$$y = r * f(Wx + b)\# \tag{4}$$

Це змушує нейромережу під час тренування випадковим чином «вимикати» частину зв'язків (наприклад, 20 %), формуючи більш стійкі та незалежні ознаки, що є критично важливим для тестування пристрою в реальних фізичних умовах [5].

На відміну від класичних повнозв'язних шарів, де кожен нейрон має власну унікальну вагу для кожної вхідної ознаки, згорткові мережі (Convolutional Neural Networks) використовують концепцію «спільних ваг» (weight sharing). Це дозволяє значно оптимізувати загальну кількість параметрів моделі, що зберігаються у Flash-пам'яті. Більше того, 1D CNN здатні ефективно враховувати локальну просторово-часову структуру сигналу. Одновимірна згортка (1D Convolution) працює як математичний "сканер", що ковзає вздовж часового ряду або спектрального масиву, виявляючи специфічні мікрорухи (наприклад, різкі прискорення або плавні нахили) незалежно від того, в який саме момент часу вони відбулися у межах ковзного вікна [6]. Загальну архітектуру такого підходу для задач розпізнавання патернів сигналів зображено на рис. 3.

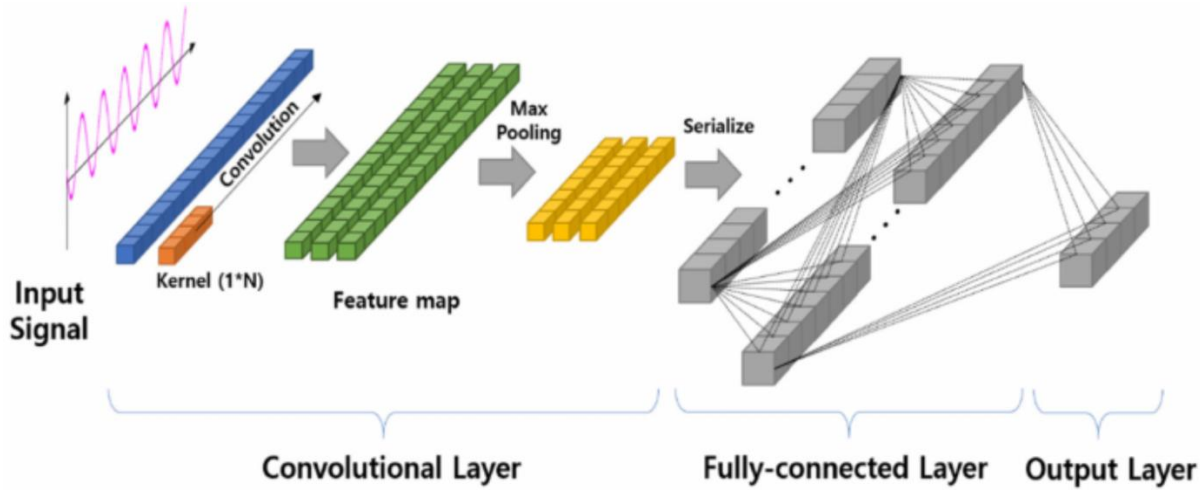

Рис. 3. Архітектура одновимірної згорткової нейромережі (1D CNN) для обробки та класифікації часових рядів [7]

Вихідне значення згортки y для позиції i обчислюється за формулою

$$y[i] = \sum_{j=0}^{k-1} w[j] * x[i + j]\# \tag{5}$$

де x – вхідний сигнал, w – ядро згортки (фільтр) розміром k.

Для використання 1D CNN у пайплайні, вхідні дані спочатку проходять через шар Reshape для набуття багатовимірної тензорної структури. Після вилучення хвиль та специфічних патернів руху згортковими фільтрами, до архітектури зазвичай інтегрується шар субдискретизації – Max Pooling (як показано на рис. 3). Цей механізм проходить по згенерованих картах ознак і залишає лише максимальні значення у заданих локальних вікнах. Це дозволяє кратно зменшити просторову розмірність даних, відкидаючи шум та зберігаючи лише найсильніші ознаки руху, що є критично важливим кроком для економії дефіцитної оперативної пам'яті (RAM) ESP32. Нарешті, стиснутий багатовимірний масив конвертується назад у плоский вектор за допомогою шару Flatten перед подачею на фінальний класифікатор. Така архітектура суттєво підвищує загальну точність системи та її стійкість до зміщення жестів у часі, але математична складність операцій згортки вимагає більших витрат процесорного часу (Inference Time).

Гібридна архітектура (CNN + Deep Dense + Dropout) концептуально об’єднує сильні сторони всіх попередніх підходів, створюючи потужний багатоетапний конвеєр обробки просторових даних. У цій топології одновимірні згорткові шари виконують роль автоматичного екстрактора ознак (Feature Extractor). Їхня задача полягає у вилученні базових мікрорухів та низькорівневих часових патернів із даних гіроскопа та акселерометра. Далі ці стиснуті багатовимірні абстракції передаються до глибокого повнозв’язного блоку, який працює як основний класифікатор, знаходячи складні нелінійні кореляції між знайденими патернами. Оскільки така багаторівнева конструкція схильна до швидкого перенавчання, на етапі прийняття фінального рішення обов’язково застосовується жорстка Dropout-регуляризація (Rate 0.25). Очікується, що цей змішаний алгоритм продемонструє найвищий показник Accuracy, наближений до абсолютного. Однак його розгортання дозволить наочно продемонструвати фундаментальний компроміс граничних обчислень (Edge AI): досягнення максимальної експлуатаційної точності неминуче вимагає найвищого споживання Flash-пам'яті, RAM та процесорного часу мікроконтролера серед усіх тестованих IoT-моделей.

3. Апаратна частина та методологія дослідження

Сімейство мікроконтролерів ESP32 від компанії Espressif Systems є стандартом у сфері Інтернету речей (IoT). Проте для задач Edge AI, де критичним фактором є оптимізація обчислювальних ресурсів та енергоспоживання автономного пристрою, класичні двоядерні моделі часто виявляються надлишковими. Тому для даного дослідження було обрано чип ESP32-C3, побудований на базі сучасної відкритої архітектури RISC-V [8]. Порівняльна характеристика базових платформ сімейства наведена в табл. 1.

Таблиця 1

Порівняння основних характеристик мікроконтролерів сімейства ESP32

Характеристика	ESP32-WROOM-32	ESP32-S3	ESP32-C3
Архітектура ядра	Двоядерний Xtensa (32-bit)	Двоядерний Xtensa (32-bit)	Одноядерний RISC-V (32-bit)
Тактова частота	До 240 МГц	До 240 МГц	До 160 МГц
Оперативна пам’ять (SRAM)	520 КБ	512 КБ	400 КБ
Бездротові інтерфейси	Wi-Fi 4, Bluetooth 4.2	Wi-Fi 4, Bluetooth 5.0	Wi-Fi 4, Bluetooth 5.0 (LE)
Енергоспоживання	Високе	Середнє	Наднизьке

Як видно з табл. 1, архітектура RISC-V в ESP32-C3 забезпечує оптимальний компроміс: чип має достатній обсяг оперативної пам'яті (400 КБ) для розгортання оптимізованих моделей TinyML при кардинальному зниженні енергоспоживання. Фізична реалізація стенда виконана на базі відлагоджувальної плати ESP32-C3 SuperMini. Її вбудований порт USB Type-C (який підтримує апаратний протокол USB Serial/JTAG) є ідеальним інструментом для швидкого прототипування, оскільки забезпечує високошвидкісний безперервний потік сирих даних безпосередньо в утиліту edge-impulse-data-forwarder.

Для збору просторової інформації у стенді використано модуль GY-521, побудований на базі мікроелектромеханічного (MEMS) чипа MPU6050 [9]. Цей сенсор об'єднує 3-осьовий акселерометр та 3-осьовий гіроскоп (загалом 6 ступенів свободи – 6-DOF). Він забезпечує високу точність вимірювань завдяки наявності незалежних 16-бітних аналого-цифровим перетворювачів (АЦП) для кожного каналу, що є критично важливим для захоплення найменших мікрорухів під час виконання жестів. Для забезпечення стабільних логічних рівнів модуль GY-521 живиться від виводу 3V3 мікроконтролера. Обмін даними здійснюється за протоколом I2C на швидкості до 400 кГц: лінію даних SDA підключено до виводу загального призначення GPIO9, а лінію тактування SCL – до виводу GPIO8. Загальну принципову схему підключення компонентів апаратного стенда наведено на рис. 4.

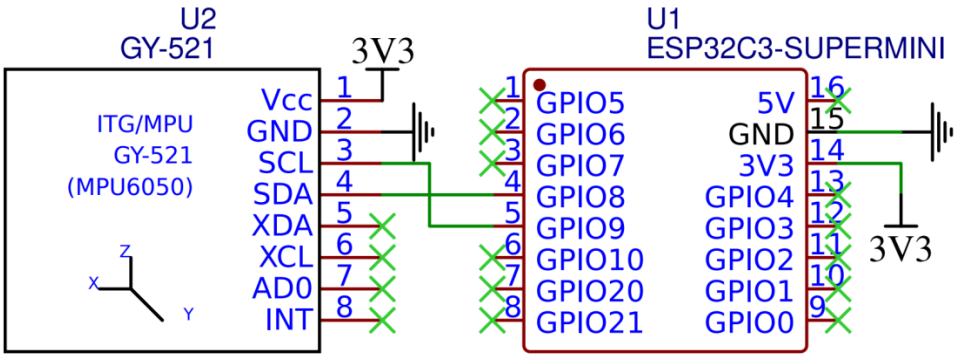

Рис. 4. Електрична принципова схема підключення сенсорного модуля GY-521 (MPU6050) до плати ESP32-C3 SuperMini

У парадигмі машинного навчання якість та репрезентативність набору даних (датасету) мають значно більший вплив на кінцеву точність моделі, ніж складність самої нейромережевої архітектури. Згідно з фундаментальними дослідженнями у сфері Data-Centric AI здатність моделі до узагальнення напряму залежить від варіативності зібраних патернів та відсутності монолітного фонового шуму [10]. Крім того, правильний збір даних для вбудованих систем вимагає жорсткої апаратної синхронізації частоти дискретизації датчика з інтервалами очікування нейромережі.

Для реєстрації просторових рухів було розроблено спеціалізоване мікропрограмне забезпечення в середовищі Arduino IDE. Щоб гарантувати ідеальну частоту збору даних у 50 Гц (опитування кожні 20 мс), алгоритм побудовано за неблокуючою архітектурою з використанням апаратного таймера мікроконтролера, що є рекомендованою практикою для систем TinyML [11]. Обмін даними з модулем MPU6050 налаштовано на підвищену частоту шини I2C (400 кГц, Fast Mode). Мікроконтролер миттєво зчитує 6 осей (3 осі акселерометра та 3 осі гіроскопа) і передає їх у вигляді відформатованого рядка через послідовний порт (Serial) на комп'ютер. На стороні ПК консольна утиліта Edge Impulse Data Forwarder автоматично перехоплює цей потік, криптографічно підписує його та безперервно маршрутизує на хмарні сервери.

Загальний обсяг зібраного набору даних становить 3 хвилини 20 секунд (20 незалежних записів по 10 секунд кожен). Методологія реєстрації багатьох коротких 10-секундних семплів, замість запису кількох суцільних довгих блоків, є критично важливою для якості навчання. Такий підхід дозволяє перехоплювати пристрій, змінювати початкове положення руки, вносити природні мікрозатримки та уникати перенавчання моделі на монотонному апаратному шумі однієї довгої сесії.

Сформований датасет містить три цільові класи рухів. Перший клас – «Idle» (7 записів), що відповідає стану спокою, коли плата лежить нерухомо або знаходиться в руці без активних рухів. Другий клас – «Shake» (7 записів), який репрезентує активне хаотичне трясіння плати у просторі. Третій клас – «Circle» (6 записів), що описує обертання плати по колу. Для досягнення максимальної стійкості моделі цей рух навмисно реєструвався з високою дисперсією: у різних площинах, під різними кутами нахилу, за та проти годинникової стрілки, а також із різною швидкістю виконання.

Для забезпечення об’єктивної оцінки алгоритмів та запобігання перенавчанню, зібраний масив було розділено на тренувальну (Training) та тестову (Test) вибірки у класичній пропорції 80/20. При цьому було застосовано строгий стратифікований поділ для збереження балансу класів. З 20 записів до тестової вибірки було ізольовано 4 семпли (рівно 20 %): два рухи «circle», один «idle» та один «shake». Тренувальні 16 семплів використовувалися виключно для налаштування вагових коефіцієнтів, тоді як тестові дані нейромережа вперше побачила лише на етапі фінальної валідації результатів.

Перед етапом машинного навчання сирі дані з акселерометра та гіроскопа проходять попередню підготовку. Для цього на платформі Edge Impulse було сформовано базовий кон-

веєр обробки. Оскільки зареєстрована частота опитування датчика становить 50 Гц, розмір ковзного вікна (Window size) було встановлено на рівні 2000 мс. Це дозволяє гарантовано захопити повний цикл будь-якого з тестованих мікрорухів. Крок зсуву вікна (Window increase) встановлено на 200 мс, що забезпечує щільне перекриття даних та штучно збільшує кількість тренувальних прикладів без необхідності фізичного збору додаткових семплів.

Наступним кроком пайплайну є блок цифрової обробки сигналів (Spectral Analysis). Замість передачі сирих часових рядів, які є надто ресурсоємними для мікроконтролера, алгоритм застосовує Швидке перетворення Фур'є (FFT) з довжиною 16 точок для кожної з 6 осей простору. Блок генерує компактний набір спектральних ознак, включаючи спектральну потужність та частотні характеристики.

Профільована продуктивність цього підготовчого DSP-блоку для мікроконтролера ESP32 становить 47 мс процесорного часу та вимагає 3 КБ пікового споживання оперативної пам'яті (Peak RAM Usage). Ці показники є фіксованими апаратними витратами для даного пайплайну. Кінцева ефективність кожної з тестованих нейромережевих архітектур буде додаватися до цих базових значень при розрахунку загального часу інференсу системи.

Для забезпечення чистоти експерименту та об'єктивності подальшого порівняльного аналізу, базові гіперпараметри навчання (Training settings) залишалися суворо незмінними для всіх чотирьох досліджуваних топологій. Відповідно до конфігурації середовища Edge Impulse процес оптимізації вагових коефіцієнтів тривав протягом 100 циклів навчання (Epochs). Розмір пакету даних для одного кроку градієнтного спуску (Batch size) становив 32 зразки, а швидкість навчання (Learning rate) була зафіксована на оптимізованому рівні 0.0005. Для постійного моніторингу процесу та уникнення перенавчання на етапі тренування 20 % тренувальної вибірки автоматично ізолювалося під внутрішню валідацію (Validation set size). Важливим аспектом конфігурації стала активація профілювання моделей у квантованому цілочисельному форматі Int8 (Profile int8 model), оскільки саме 8-бітна арифметика є цільовою для розгортання на апаратній базі ESP32. Вхідний шар (Input layer) для всіх архітектур жорстко приймав 78 спектральних ознак, згенерованих попереднім блоком цифрової обробки сигналів. За цих ідентичних умов було послідовно синтезовано та навчено чотири нейромережеві моделі.

Перша базова повнозв'язна модель (Shallow Dense) була сконфігурована як мінімалістичний класифікатор. Її архітектура безпосередньо після вхідного шару включала лише два приховані шари типу Dense на 20 та 10 штучних нейронів відповідно, після чого багатовимірний простір звужувався до вихідного шару для розподілу на три цільові класи жестів.

Друга глибока повнозв'язна мережа з регуляризацією (Deep Dense + Dropout) відрізнялася розширеною обчислювальною ємністю. Після вхідного блоку розташовувався шар Dense на 32 нейрони, за яким слідував шар просторового відсіву Dropout із коефіцієнтом 0,2 (примусове випадкове вимикання 20 % нейронних зв'язків у кожній епосі). Далі оптимізовані дані передавалися на наступний глибокий Dense-шар із 16 нейронами та фінальний класифікатор.

Третя топологія, одновимірна згорткова мережа (1D CNN), вимагала суворого попереднього узгодження розмірностей тензора між блоком вилучення ознак та згортковим шаром. Оскільки апарат математичної згортки (1D Conv) концептуально функціонує виключно з тривимірними масивами даних формату «пакет-кроки-канали», безпосередньо після вхідного шару було інтегровано шар просторового переформатування (Reshape). Його завдання полягало в перетворенні плоского вхідного вектора з 78 спектральних ознак у структуру матриці-стовпця (78 columns), штучно додаючи необхідний вимір каналу. Основна математична обробка виконувалася комбінованим шаром одновимірної згортки та субдискретизації (1D Conv / Pool layer) з використанням 8 незалежних фільтрів та розміром ядра згортки (kernel size), що дорівнює 3. Після вилучення локальних просторово-часових патернів та паралельного зменшення розмірності за допомогою пулінгу, сформований багатовимірний тензор вирівнювався в лінійний масив за допомогою шару Flatten. На фінальному етапі ці підго-

товлені високорівневі ознаки передавалися на базовий повнозв'язний шар (Dense layer) із 16 нейронами для здійснення остаточної класифікації.

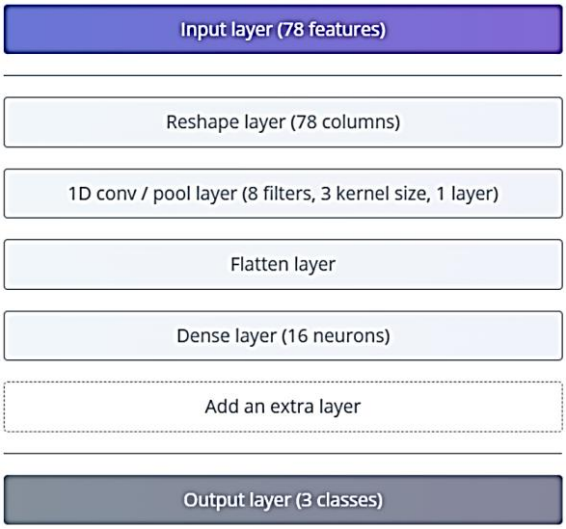

Рис. 5. Архітектура шарів одновимірної згорткової нейронної мережі (1D CNN)

Четверта гібридна архітектура (CNN + Deep Dense + Dropout) являла собою найбільш комплексний та ресурсоємний математичний конвеєр у даному дослідженні. Як і в попередній топології, вхідні дані спочатку проходили через шар Reshape для набуття правильної тензорної структури. Однак далі вони оброблялися посиленим згортковим шаром із розширеною вдвічі кількістю фільтрів (16 одиниць) при незмінному розмірі ядра 3. Оскільки збільшення кількості фільтрів пропорційно збільшує обсяг згенерованих ознак і підвищує ризик перенавчання (overfitting), для стабілізації цього масиву безпосередньо після згортки застосовувався шар просторового відсіву (Dropout). Відповідно до конфігурації моделі, коефіцієнт відсіву (rate) становив 0.2, що забезпечувало випадкове вимикання 20 % зв’язків під час кожної епохи навчання. Після зведення тензора в плоский вектор шаром Flatten, ознаки передавалися до глибокого повнозв’язного шару на 16 нейронів, який діяв як головний класифікатор перед вихідним шаром нейромережі.

4. Порівняльний аналіз продуктивності нейромережевих архітектур

Усі розроблені моделі були натреновані в ідентичних умовах базового середовища, після чого були отримані наступні результати апаратного профілювання. Для кожної з чотирьох топологій компілятор Edge Impulse згенерував дві версії кінцевої моделі: неоптимізовану з 32-бітною точністю з плаваючою комою (Float32) та квантовану цілочисельну версію (Int8), спеціально адаптовану для роботи на мікроконтролерах. Зведені емпіричні метрики точності класифікації та споживання апаратних ресурсів платформи ESP32-C3 наведено в табл. 2.

Насамперед спостерігається яскраво виражена деградація точності класифікації при застосуванні 8-бітного квантування (Int8) для всіх без винятку моделей. У повнозв’язних архітектурах точність різко падає з референсних 97–100 % до неприйнятних 59–60 %. Цей феномен пояснюється тим, що згенеровані спектральні ознаки, які є результатом роботи алгоритму швидкого перетворення Фур’є, містять критично важливі тонкі дробові значення. Примусове стиснення цих вагових коефіцієнтів у вузький 8-бітний діапазон призводить до згортування математичного простору ознак, через що нейромережа втрачає здатність розрізняти схожі мікрорухи у просторі. Цікавим спостереженням є те, що квантування для одновимірної згорткової моделі (1D CNN) спрацювало дещо краще, зберігши точність на рівні 78.0 %. Це зумовлено здатністю згортки вилучати більш грубі та стійкі просторові патерни, проте цей показник все одно залишається недостатнім для надійного використання пристрою в реальних умовах.

Таблиця 2

Зведені результати апаратного профілювання нейромережевих архітектур

Архітектура	Тип компіляції	Точність, %	Час інференсу, мс	Peak RAM, КБ	Flash Usage, КБ
Shallow Dense	Float32	97.7	9 с	1.6	17.7
Shallow Dense	Int8	59.8	2 с	1.4	15.8
Deep Dense + Dropout	Float32	100.0	5 с	1.7	22.7
Deep Dense + Dropout	Int8	60.6	3 с	1.5	17.1
1D CNN	Float32	100.0	17 с	2.1	29.5
1D CNN	Int8	78.0	2	3.6	30.6
Гібридна (CNN + Dense)	Float32	98.5	32	2.1	37.3
Гібридна (CNN + Dense)	Int8	59.1	2	3.7	32.7

Разом з тим, неоптимізовані моделі (Float32) демонструють видатні результати і є найбільш збалансованим рішенням для поставленої задачі. Як глибока мережа з регуляризацією (Deep Dense + Dropout), так і одновимірна згорткова мережа (1D CNN) досягли абсолютної точності на валідаційній вибірці (100.0 %). Гібридна архітектура також продемонструвала надзвичайно високу точність (98.5 %), успішно узагальнивши складні рухи. Це доводить, що як правильно налаштована Dropout-регуляризація в Dense-мережах, так і застосування просторових згорткових фільтрів дозволяють ефективно боротися з апаратним шумом MEMS-гіроскопа та уникати перенавчання.

Оцінюючи апаратні витрати, варто відзначити, що використання форматів Float32 є абсолютно виправданим для мікроконтролерів класу ESP32-C3. Найважча з протестованих моделей (Гібридна Float32) споживає лише 2.1 КБ оперативної пам'яті та 37.3 КБ Flash-пам’яті, що становить мізерну частку від загальних апаратних ресурсів чипа. З точки зору процесорного часу, повнозв’язні моделі виконують інференс найшвидше (5–9 мс), тоді як 1D CNN потребує 17 мс, а гібридна архітектура через свою математичну комплексність – 32 мс. Однак, навіть у найскладнішому розрахунковому сценарії з використанням гібридної мережі, сумарний час реакції системи, що складається з роботи блоку цифрової обробки сигналів (47 мс) та інференсу самої нейромережі (32 мс), не перевищує 79 мс. Для найуспішнішої моделі 1D CNN цей показник становить 64 мс. Оскільки отримані значення з величезним запасом вписуються у жорсткі вимоги реального часу для систем розпізнавання жестів (відгук до 100 мс), процес квантування моделей (Int8) у даному конкретному дослідженні є об’єктивно недоцільним через катастрофічну втрату експлуатаційної точності заради економії кількох мілісекунд.

Висновки

Проведено комплексний порівняльний аналіз ефективності розгортання моделей машинного навчання (TinyML) для задачі класифікації просторових сенсорних даних на базі мікроконтролера ESP32-C3. На основі результатів апаратного профілювання чотирьох різних нейромережевих топологій (Shallow Dense, Deep Dense + Dropout, 1D CNN та гібридної моделі) можна зробити низку важливих науково-практичних висновків.

По-перше, доведено, що одновимірна згорткова мережа (1D CNN) є найбільш збалансованим та оптимальним рішенням для розпізнавання патернів руху. Завдяки здатності згорткових фільтрів ефективно вилучати просторово-часові ознаки, ця модель у форматі Float32 досягла абсолютної точності (100.0 %) на валідаційній вибірці. Натомість гібридна архітектура, хоч і продемонструвала високу точність (98.5 %), виявилася надлишковою через більшу математичну комплексність та збільшений час інференсу.

По-друге, найважливішим емпіричним спостереженням роботи є виявлена критична деградація точності класифікації при застосуванні 8-бітного квантування (Int8). Оскільки спектральні ознаки, отримані в результаті швидкого перетворення Фур’є (FFT), містять життєво важливі тонкі дробові значення, їх примусове стиснення у цілочисельний формат призводить до руйнування математичного простору ознак. Як наслідок, точність моделей падає до неприйнятних 59–78 %, що робить процес квантування для подібних завдань об’єктивно недоцільним.

По-третє, аналіз апаратних витрат підтвердив, що використання моделей із 32-бітною точністю (Float32) є абсолютно виправданим та безпечним для сучасних мікроконтролерів класу ESP32-C3. Навіть найважча топологія (Гібридна Float32) споживає лише мізерну частку ресурсів: 2.1 КБ оперативної та 37.3 КБ Flash-пам'яті.

Загальний час реакції розробленої системи (від захоплення сирих даних до видачі результату), що складається з роботи блоку цифрової обробки сигналів (47 мс) та інференсу найуспішнішої моделі 1D CNN (17 мс), становить 64 мс. Ці показники з величезним запасом вписуються у жорсткі вимоги систем реального часу (відгук до 100 мс). Таким чином, розроблений підхід дозволяє створювати високоефективні, точні та автономні вбудовані пристрої на базі Edge AI, здатні аналізувати складні просторові рухи без залучення хмарних обчислювальних потужностей.

Список літератури:

1. Banos O., Galvez J. M., Damas M., Pomares H., & Rojas I. (2014). Window size impact in human activity recognition // Sensors. 2014. No14(4). P. 6474–6499. URL: https://www.mdpi.com/1424-8220/14/4/6474.

2. Oppenheim A. V., & Schafer R. W. (2010). Discrete-Time Signal Processing (3rd ed.). Pearson.

3. Goodfellow I., Bengio Y., & Courville A. (2016). Deep Learning. MIT Press. URL:https://www.deeplearningbook.org/.

4. Jacob B., Kligys S., Chen B., Zhu M., Tang M., Howard A., Adam H., & Kalenichenko D. Quantization and training of neural networks for efficient integer-arithmetic-only inference // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018. P. 2704–2713. URL:https://openaccess.thecvf.com/content_cvpr_2018/html/Jacob_Quantization_and_Training_CVPR_2018_paper.html.

5. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., & Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting // The Journal of Machine Learning Research. 2014. No15(1). P.1929–1958. URL:https://jmlr.org/papers/v15/srivastava14a.html.

6. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. 1D convolutional neural networks and applications: A survey // Mechanical Systems and Signal Processing. 2021. No 151. P. 107398. URL:https://doi.org/10.1016/j.ymssp.2020.107398.

7. Kim J., Kim H., & Kim J. Hyperparameter Optimization Method Based on Harmony Search Algorithm to Improve Performance of 1D CNN Human Respiration Pattern Recognition System // Sensors. (2020). Vol.20(13). P. 3697. URL:https://www.mdpi.com/1424-8220/20/13/3697.

8. Espressif Systems. (2024). ESP32-C3 Series Datasheet. URL:https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf.

9. InvenSense. (2013). MPU-6000 and MPU-6050 Product Specification Revision 3.4. URL:https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf.

10. Roh Y., Heo G., & Whang,S. E. A survey on data collection for machine learning: A big data-ai integration perspective // IEEE Transactions on Knowledge and Data Engineering. 2019. Vol. 33(4). P. 1328–1347. URL:https://ieeexplore.ieee.org/document/8936454.

11. Warden P., & Situnayake D. (2019). TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly Media. URL:https://www.oreilly.com/library/view/tinyml/9781492052001/.

Надійшла до редколегії 07.01.2026
Прийнята до друку після рецензування 23.04.2026
Публікація (оприлюднення) 30.04.2026

Відомості про авторів:

Погуляй Данило Олегович – Харківський національний університет радіоелектроніки, студент кафедри комп'ютерної інженерії та управління; Україна; e-mail: danylo.pohuliai@nure.ua; ORCID: https://orcid.org/0009-0004-4789-9858

Голубничий Дмитро Юрійович – канд. техн. наук, доцент, Харківський національний університет радіоелектроніки, Харківський національний економічний університет імені Семена Кузнеця, Україна; e-mail: dmytro.holubnychyi1@nure.ua; ORCID: https://orcid.org/0000-0002-6873-7004

Єсіна Марина Віталіївна – канд. техн. наук, доцент, Харківський національний університет імені В. Н. Каразіна, завідувачка кафедри кібербезпеки інформаційних систем, мереж і технологій, навчально-науковий інститут комп'ютерних наук та штучного інтелекту; АТ Інститут Інформаційних Технологій", науковий співробітник-консультант; Україна; e-mail: m.v.yesina@karazin.ua; ORCID: https://orcid.org/0000-0002-1252-7606