

# МОДЕЛЮВАННЯ БАГАТОКЛАСОВОЇ КЛАСИФІКАЦІЇ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ PYTORCH

ШВЕЦЬ В. О., 1 курс, 6м група ФІТ ДТЕУ,  
спеціальність «Комп'ютерні науки»,  
освітня програма «Комп'ютерні науки»

*У статті здійснено теоретичний аналіз можливостей фреймворку PyTorch у контексті задач багатокласової класифікації. Розкрито переваги його використання для побудови ефективних моделей глибокого навчання, зокрема в аспектах архітектурної гнучкості, налаштування гіперпараметрів і реалізації процесів оптимізації. Також подано порівняльну оцінку з іншими поширеними фреймворками, що дозволяє виявити специфічні риси PyTorch, важливі для дослідницької та прикладної діяльності.*

*The article provides a theoretical analysis of the capabilities of the PyTorch framework in the context of multiclass classification tasks. It highlights the advantages of using PyTorch for building effective deep learning models, particularly in terms of architectural flexibility, hyperparameter tuning, and optimization processes. A comparative evaluation with other widely used frameworks is also presented, allowing the identification of specific features of PyTorch that are essential for both research and practical applications.*

**Актуальність.** У сучасних умовах стрімкого розвитку технологій штучного інтелекту та зростаючих обсягів даних завдання багатокласової класифікації набуває особливої актуальності. Такий тип класифікації є широко застосовуваним у практиці аналізу даних, зокрема у сферах обробки зображень, текстів, розпізнавання об'єктів, медичних діагнозів тощо. З огляду на складність подібних завдань, необхідним є використання інструментів, що забезпечують гнучкість у побудові моделей та ефективність у їх тренуванні. Одним із таких інструментів є фреймворк PyTorch, який останніми роками набуває популярності серед дослідників і практиків завдяки своїй модульності, високому рівню контрольованості процесів навчання та активній підтримці з боку спільноти розробників.

**Метою** цього дослідження є вивчення можливостей PyTorch для моделювання багатокласової класифікації, зокрема виявлення його переваг і обмежень при вирішенні таких завдань.

**Завданням дослідження** є теоретичне узагальнення ключових підходів, методів і технічних аспектів, що пов'язані із застосуванням PyTorch у таких задачах. Аналіз включає порівняння фреймворку PyTorch з іншими популярними інструментами, такими як TensorFlow та Keras, з точки зору їх придатності для вирішення проблем багатокласової класифікації.

**Об'єктом** виступає процес нейронної мережі для багатокласової класифікації з використанням фреймворка PyTorch

**Предмет дослідження** охоплює архітектури нейронних мереж, механізми оптимізації, вибір функцій втрат і активацій, а також загальні технічні аспекти, що впливають на ефективність роботи моделей у PyTorch.

У задачах багатокласової класифікації, які передбачають розподіл вхідних даних на три і більше класів, вирішальне значення має вдало підібрана архітектура моделі. Адже саме вона визначає, наскільки ефективно система зможе врахувати особливості даних і відокремити кожен клас. Ключовими етапами в побудові такої моделі є підготовка даних, вибір функцій втрат (наприклад, cross-entropy loss), оптимізаційного алгоритму, а також налаштування гіперпараметрів. Важливо також не забувати про регуляризацию, яка допомагає уникнути перенавчання та покращити узагальнювальні здібності моделі.

У цьому контексті PyTorch постає не просто як інструмент, а як повноцінне середовище для наукових досліджень і реалізації складних нейронних архітектур. За визначенням його



розробників, PyTorch – це науковий обчислювальний інструментарій і платформа для глибокого навчання, що поєднує гнучкість, високу продуктивність і зручність у використанні [1]. Центральною ланкою цієї платформи є тензорна бібліотека, яка забезпечує широкі можливості для обробки багатовимірних даних і реалізації складних математичних операцій, необхідних для сучасного машинного навчання.

Серед ключових переваг PyTorch слід відзначити гнучку архітектуру, що дозволяє динамічно змінювати структуру нейронних мереж без необхідності переписувати весь код. Завдяки підтримці Eager execution (динамічної побудови обчислювального графа) фреймворк забезпечує зручність при налагодженні та оптимізації моделей, адже зміни в архітектурі можуть бути перевірені в реальному часі. Це значно спрощує процес розробки, особливо коли йдеться про експериментальні та наукові розробки. Крім того, цей фреймворк дозволяє працювати безпосередньо з тензорами – багатовимірними масивами даних, що є основою для побудови складних моделей машинного навчання. Завдяки інтеграції з CUDA (платформою для обчислень на GPU) PyTorch забезпечує високу продуктивність, що є особливо важливим при роботі з великими обсягами даних.

Основною перевагою PyTorch є його наближеність до «рідної» логіки Python, що робить фреймворк інтуїтивно зрозумілим для розробників з різним рівнем підготовки. В основі PyTorch лежить структура тензора – багатовимірного масиву, аналогічного NumPy, але з розширеними можливостями для виконання обчислень на GPU. Така структура дозволяє будувати складні математичні моделі та забезпечує високу продуктивність у процесі навчання мереж, включно з можливістю масштабування на паралельні обчислювальні середовища [2].

Водночас, варто враховувати, що ефективне застосування PyTorch у задачах багатокласової класифікації потребує певного рівня підготовки з боку користувача. Зокрема, необхідне розуміння принципів роботи нейронних мереж, методів регуляризації, особливостей ініціалізації ваг та вибору гіперпараметрів. Правильна побудова конвеєра навчання та тестування моделі також є важливою складовою загального успіху.

Для ілюстрації переваг PyTorch у контексті багатокласової класифікації, нижче наведено порівняльну таблицю з TensorFlow:

*Таблиця*

### **Порівняння PyTorch та TensorFlow у контексті задач багатокласової класифікації**

Характеристика	PyTorch	TensorFlow
Побудова обчислювального графа	Динамічна (Eager execution)	Статична (Graph mode)
Зручність у відлагодженні	Висока, інтегрується зі стандартним debugger	Обмежена у Graph mode
Популярність у дослідницькому середовищі	Висока	Вища в індустрії
Модульність	Гнучка архітектура, можливість створення власних шарів	Просте API, менш гнучке
Підтримка багатокласових задач	Повноцінна, з великою кількістю прикладів	Також повна, інтегрована в екосистему

Як показує наведене порівняння, PyTorch надає більшу гнучкість у контексті дослідницьких розробок, тоді як TensorFlow більше орієнтований на використання в промислових рішеннях. Однак обидва фреймворки мають потужний функціонал для реалізації багатокласової класифікації, і вибір між ними залежить від конкретних потреб дослідника або розробника.

**Висновки.** Використання фреймворку PyTorch для побудови моделей багатокласової класифікації не лише відкриває нові горизонти для досліджень, але й стимулює розвиток нових підходів у галузі машинного навчання. Гнучкість та можливість швидко адаптувати моделі до різних умов роблять PyTorch важливим інструментом для науковців і практиків. Цей фреймворк дає змогу не лише теоретично досліджувати нейронні мережі, а й ефективно експериментувати з різними архітектурами і техніками. Завдяки цьому відкриваються нові можливості для глибшого розуміння складних моделей класифікації, що може призвести до розробки більш ефективних алгоритмів і технологій у майбутньому. Враховуючи цей потенціал, подальші дослідження та порівняння PyTorch з іншими платформами дозволять знайти оптимальні рішення для конкретних задач і сприятимуть розвитку галузі в цілому.

### Список використаних джерел

1. PyTorch // ITWiki. URL: <https://itwiki.dev/data-science/ml-reference/ml-glossary/pytorch> (дата звернення: 10.04.2025).
2. Stevens E., Antiga K. L., Viehmann T. Deep Learning with PyTorch. – San Francisco: O’Reilly Media, 2020. – 6 р.

Роботу виконано під науковим керівництвом канд. фіз.-мат. наук, доцента  
ФІЛІМОНОВОЇ Т. О.

## СИМУЛЯТОР МОРСЬКОГО ДРОНА – МІННОГО ЗАГОРОДЖУВАЧА НА ОСНОВІ РУШІЯ UNREAL ENGINE 5

ШЕВЧЕНКО Д. М., 1 курс, 6м група ФІТ ДТЕУ,  
спеціальність «Комп’ютерні науки»,  
освітня програма «Комп’ютерні науки»

*У статті проаналізовано процес розробки гри з використанням програмного забезпечення Unreal Engine, що використовується для створення описаної послідовної роботи процесів розробки ігрового контенту.*

*The article analyzes the game development process using the Unreal Engine software, which is used to create the described sequential work of game content development processes.*

**Актуальність:** Сучасні військові технології активно використовують симулятори для навчання, планування та відпрацювання бойових сценаріїв. Особливо це стосується дистанційно керованих систем – безпілотників, у тому числі морських дронів. Симуляція дозволяє моделювати дії морських апаратів у реальному або умовно-реальному середовищі без ризику для обладнання чи персоналу.

У цій статті розглядається процес створення симулятора морського дрона – мінного загороджувача з використанням сучасного ігрового рушія Unreal Engine 5. Розробка такого симулятора дозволяє вивчити можливості моделювання водного середовища, фізики дрона, мінування та навігації, а також перспективи використання ігрових рушіїв у військових та навчальних цілях.

**Мета:** Симулятор створюється з кількома ключовими цілями:

1. Відпрацювання сценаріїв мінного загородження у прибережній зоні або на відкритій воді;

