



## АНОТАЦІЯ

**Козачок В.О. Розробка методів глибокого машинного навчання для середовища MATLAB у розв'язанні задач навігації. – Рукопис.**

Кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю 121 – Інженерія програмного забезпечення. – Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПм-61 // м.Тернопіль, 2020 // С., рис. – 25 , табл. – 0 , додат. – 3 , бібліогр. – 12

Мета кваліфікаційної роботи полягає в використанні сучасних інформаційних технологій для розробки методів глибокого машинного навчання, з врахуванням специфіки конкретної галузі і підвищенням конкурентоспроможності продукту за рахунок впровадження інноваційних підходів до роботи з машинним навчанням.

Практичне застосування – розроблено надійний програмний продукт, що дозволить підвищити ефективність та продуктивність роботи та який є новим, унікальним та відповідає поставленій задачі.

Технічні вимоги – методи розробки базуються на технології та високорівневій мові програмування MATLAB.

**Ключові слова:** МАШИННЕ НАВЧАННЯ, MATLAB, НЕЙРОН, ГРАФІК.

## ABSTRACT

**Kozachok V. Development of deep machine learning methods for MATLAB environment in solving navigation problems. – Manuscript.**

The master degree thesis for the qualification level of magistr in the specialty 121 — Software Engineering. – Ternopil Ivan Pul'ui National Technical University, Faculty of Computer Information Systems and Software Engineering, Software Engineering Department, group SPm-61 // Ternopil, 2020 //

Pages. – 75 , pictures. – 25 , tables. – 0 , supp. – 3 , bibl.ref. – 12

The purpose of the qualification work is to use modern information technologies to develop methods of deep machine learning, taking into account the specifics of a particular industry and increase the competitiveness of the product through the introduction of innovative approaches to working with machine learning.

Practical application - a reliable software product has been developed that will increase the efficiency and productivity of work and which is new, unique and meets the task.

Technical requirements - development methods are based on MATLAB technology and high-level programming language.

**Keywords:** MACHINE LEARNING, MATLAB, NEURON, GRAPH.

## ЗМІСТ

АНОТАЦІЯ .....	2
ABSTRACT .....	3
ВСТУП .....	6
1 АНАЛІТИЧНИЙ ОГЛЯД В ОБЛАСТІ ДОСЛІДЖЕНЬ. ....	8
1.1 Аналіз та огляд предметної області.....	9
1.2 Основні алгоритми та принципи роботи нейронних мереж .....	15
2 ОБҐРУНТУВАННЯ ВИБОРУ НАПРЯМКУ ДОСЛІДЖЕНЬ .....	27
2.1 Вибір технологій розробки .....	27
2.2 Обґрунтування вибору середовища розробки програмної системи .....	29
3 РОЗРОБКА СКЛАДОВИХ ПРОГРАМНОГО КОМПЛЕКСУ .....	33
3.1 Моделювання літального апарату .....	33
3.2 Створення моделі місцевості .....	38
3.3 Створення моделі камери.....	41
3.4 Побудова траєкторії на зображенні .....	44
3.5 Створення тестових зображень.....	45
3.6 Тренування і тестування .....	47
3.7 Симуляція.....	49
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ..	52
4.1 Охорона праці .....	52
4.2 Забезпечення безпеки життєдіяльності при роботі з ПК.....	55
4.2.1 Параметри робочого місця .....	57

4.2.2 Вимоги до освітленості і повітряного середовища в робочій зоні.....	60
4.2.3 Допустимі рівні звуку на робочих місцях .....	62
ВИСНОВОК.....	64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	65
ДОДАТКИ.....	67

## ВСТУП

Говорячи про новітні технології, можна з упевненістю сказати, що машинне навчання та штучний інтелект дедалі більше входять у всі сфери людського життя та спрощують багато складних процесів. Штучний інтелект та машинне навчання прискорюють роботу компаній, роблять їх ефективнішими, потужнішими, виводять на новий рівень. За допомогою цих інструментів менеджери можуть вирішувати багатовимірні проблеми, що вимагають нетрадиційних підходів. Витрати можна оптимізувати за допомогою роботизованої автоматизації процесів. Ці інструменти роблять нові бізнес-моделі можливими завдяки автоматизації завдань за допомогою комп'ютерного зору, обробки природної мови та розпізнавання мови. Штучний інтелект та машинне навчання допомагають автоматизувати системні операції, контролювати ефективність бізнесу та швидше виявляти шахрайство.

Часто виникає ситуація, коли необхідна оптимізація процесів у різних сферах діяльності, при чому велика частка залежить від ресурсної складової. У таких випадках перекладання задач з обрахунку на системи машинного навчання є доцільним для перерозподілу та оптимізації ресурсів. Вруховуючи швидкість розвитку технологій машинного навчання, все складніші завдання можуть виконуватись за допомогою методів глибокого машинного навчання.

Той факт, що глибоке навчання є різновидом машинного навчання, дуже важливий, саме тому потрібно встановити відношення між машинним навчанням та глибоким навчанням. Глибоке навчання нещодавно було в центрі уваги, оскільки воно кваліфіковано вирішило деякі проблеми, які кинули виклик штучному інтелекту. Його продуктивність, безумовно, є винятковою у багатьох сферах. Однак він також стикається з обмеженнями. Як тип машинного навчання, глибоке навчання не може уникнути основних проблем, з якими стикається машинне навчання.

Для реалізації ідеї, з врахуванням особливостей предметної області та коцепції машинного навчання в цілому, ввикористано технологію та однойменну високорівневу мову програмування MATLAB. Технологія має спеціально призначені компоненти для використання в розробці методів глибокого машинного навчання, що спрощує завдання для розробника і дозволяє сконцентруватись на програмному коді та алгоритмах.

Важливим моментом є те, що тема є відносно новою і перспективною, що дозволяє застосувати максимальну кількість доступних інноваційних технологій та мати змогу впровадити систему у дію та доповнювати її відносно потреб у обраній сфері.

Об'єктом дослідження обрано актуальні питання обраної галузі, що дозволяє докладно вивчити проблематику, проаналізувати модливі способи вирішення з використанням передових методів програмування.

Предметом дослідження зазначено програмну систему на мові MATLAB, що являє собою реалізацію методів глибокого машинного навчання для вирішення проблем з визначеної предметної області. Програмна система розроблена з врахуванням вимог швидкодії та максимальної продуктивності для вирішення задач у повному обсязі.

Як наукову новизну слід розглядати використання сучасних технологій машинного навчання для розробки унікального нового програмного забезпечення з урахуванням вимог та специфіки конкретної галузі.

Застосовувана сторона результатів наукових досліджень може бути конкретною діяльністю, і продукт повинен вдосконалюватись та адаптуватись до сучасних вимог.

Виконання і оформлення роботи буде виконуватися згідно всіх стандартів та вимог, визначених методичними рекомендаціями та Державним стандартом України по відношенню до термінів і визначень, оформлення звітів та специфічної документації.

## 1 АНАЛІТИЧНИЙ ОГЛЯД В ОБЛАСТІ ДОСЛІДЖЕНЬ.

Можна легко знайти приклади, коли поняття машинного навчання та глибокого навчання використовуються як взаємозамінні в засобах масової інформації. Однак фахівці їх загалом розрізняють. Загалом, штучний інтелект, машинне навчання та глибоке навчання пов'язані таким чином: „Глибоке навчання - це різновид машинного навчання, а машинне навчання - це такий собі штучний інтелект”. Ця класифікація може бути не такою абсолютною, як закони природи, але вона є загальноновизнаною. Штучний інтелект [1] - дуже поширене слово, яке може означати багато різних речей. Це може вказувати на будь-яку форму технології, яка включає деякі інтелектуальні аспекти, а не визначити конкретну галузь технології. На відміну від цього, машинне навчання відноситься до певної галузі. Іншими словами, ми використовуємо машинне навчання для позначення конкретної технологічної групи штучного інтелекту. Саме машинне навчання також включає багато технологій. Одним з них є глибоке навчання, що є предметом цієї книги. Той факт, що глибоке навчання є різновидом машинного навчання, дуже важливий, і саме тому ми проходимо цей довгий огляд того, як пов'язані між собою штучний інтелект, машинне навчання та глибоке навчання. Глибоке навчання нещодавно було в центрі уваги, оскільки воно кваліфіковано вирішило деякі проблеми, які кинули виклик штучному інтелекту. Його продуктивність, безумовно, є винятковою у багатьох сферах. Однак він також стикається з обмеженнями. Обмеження глибокого навчання впливають з його фундаментальних концепцій, які успадкували від його предку, машинного навчання. Як тип машинного навчання, глибоке навчання не може уникнути основних проблем, з якими стикається машинне навчання. Ось чому нам потрібно переглянути машинне навчання, перш ніж обговорювати концепцію глибокого навчання.



## 1.1 Аналіз та огляд предметної області

Вибраною темою магістерської роботи є «Розробка методів глибокого машинного навчання для середовища MATLAB у розв'язанні задач навігації». Тема є досить актуальною по тій причині, що у світі розвитку новітніх технологій проблема оптимізації роботи навігаційних систем є досить актуальною у багатьох сферах застосування.

Машинне навчання - це техніка моделювання, яка включає дані. Це визначення може бути занадто коротким для початківців, щоб зрозуміти, що воно означає. Отже, дозвольте мені трохи детальніше розробити це. Машинне навчання - це техніка, яка визначає „модель” із „даних”. Тут дані буквально означають таку інформацію, як документи, аудіо, зображення тощо. „Модель” є кінцевим продуктом машинного навчання. Перш ніж ми підемо далі в модель, дозвольте мені трохи відхилитися. Чи не дивно, що визначення машинного навчання стосується лише понять даних і моделі і не має нічого спільного з "навчанням"? Сама назва відображає, що техніка аналізує дані і знаходить модель сама, а не за допомогою людини. Ми називаємо це «навчанням», оскільки процес нагадує навчання з даними для вирішення проблеми пошуку моделі. Тому дані, які машинне навчання використовує в процесі моделювання, називаються „навчальними” даними.

Насправді модель - це не що інше, як те, чого ми хочемо досягти як кінцевий продукт. Наприклад, якщо ми розробляємо систему автоматичного фільтрування для видалення спаму, фільтр спаму - це модель, про яку ми говоримо. У цьому сенсі ми можемо сказати, що модель - це те, що ми насправді використовуємо. Деякі називають модель гіпотезою. Цей термін здається більш інтуїтивним для тих, хто має статистичний досвід. Машинне навчання [2] - не єдина техніка моделювання. У галузі динаміки люди вже давно використовують певний прийом моделювання, який використовує закони Ньютона і описує рух об'єктів як ряд

рівнянь, які називаються рівняннями руху. У галузі штучного інтелекту ми маємо експертну систему, яка є моделлю вирішення проблем, що базується на знаннях та ноу-хау експертів. Модель працює так само, як і самі експерти.

Однак є деякі сфери, де закони та логічні міркування не дуже корисні для моделювання. Типові проблеми можна знайти там, де задіяний інтелект, такі як розпізнавання зображень, розпізнавання мови та обробка природної мови. Машинне навчання створено для вирішення проблем, для яких аналітичні моделі навряд чи доступні. Основною ідеєю машинного навчання є досягнення моделі з використанням навчальних даних, коли рівняння та закони не є перспективними.

Глибоке навчання [3] - це підмножина машинного навчання, яка сама по собі є підмножиною штучного інтелекту та статистики. Дослідження штучного інтелекту розпочалися незабаром після Другої світової війни [24]. Ранні роботи базувались на знанні будови мозку, логічної пропозиції та теорії обчислень Тьюрінга. Уоррен Маккалок і Уолтер Пітс створили математичну формулу для нейронних мереж на основі порогової логіки. Це дозволило дослідження нейронних мереж розділитися на два підходи: один зосереджений на біологічних процесах мозку, а інший - на застосуванні нейронних мереж до штучного інтелекту. Було продемонстровано, що будь-яка функція може бути реалізована за допомогою набору таких нейронів і що нейронна мережа може вчитися. У 1948 р. Була опублікована книга Норберта Вінера «Кібернетика», де були описані концепції управління, зв'язку та статистичної обробки сигналів. Наступним важливим кроком у нейронних мережах стала книга Дональда Хебба в 1949 р. «Організація поведінки», яка пов'язує зв'язок із навчанням у мозку. Його книга стала джерелом навчальних та адаптивних систем. Марвін Мінський та Дін Едмондс побудували перший нейронний комп'ютер у Гарварді в 1950 році. Перші комп'ютерні програми, і переважна більшість зараз, мають знання, вбудовані в код програмістом. Програміст може використовувати величезні бази даних. Наприклад, модель літального апарату може використовувати багатовимірні таблиці аеродинамічних коефіцієнтів. Таким чином, отримане програмне

забезпечення знає багато про літаки, і запуснені імітації моделей можуть подарувати програмісту та користувачам сюрпризи. Тим не менше, програмні відносини між даними та алгоритмами визначаються кодом.

У машинному навчанні взаємозв'язок між даними формується системою навчання. Дані вводяться разом із результатами, пов'язаними з даними. Це системна підготовка. Система машинного навчання пов'язує дані з результатами та придумує правила, які стають частиною системи. Коли вводяться нові дані, вони можуть дати нові результати, які не були частиною навчального набору.

Глибоке навчання відноситься до нейронних мереж з більш ніж одним шаром нейронів [4]. Назва «глибоке навчання» передбачає щось більш глибоке, і в популярній літературі прийнято стверджувати, що система навчання є «глибоким мислителем». На рисунку 1.1 показана одношарова та багатошарова мережа.

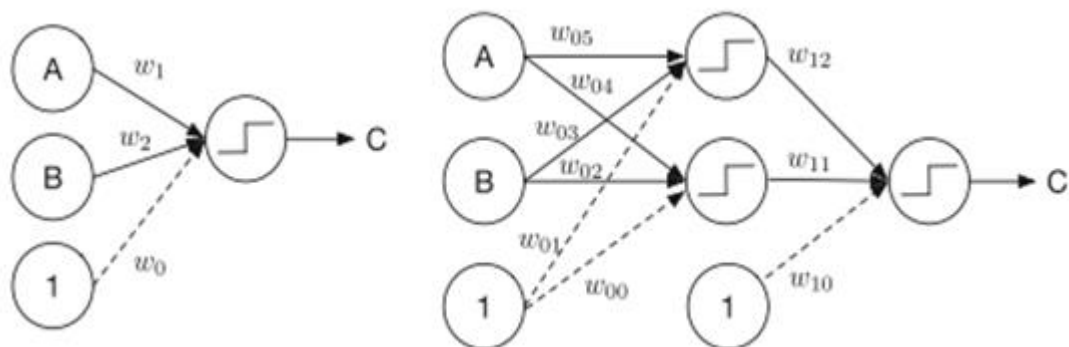


Рисунок 1.1 Одношарова та багатошарова мережі

Виявляється, багатошарові мережі можуть навчитися речам, яких одношарові мережі не можуть. Елементами мережі є вузли, де сигнали поєднуються, ваги та упередження. Зміни додаються у вузлах. В одному шарі входні дані помножуються на ваги, потім складаються в кінці, після проходження через порогову функцію. У багатошаровій мережі [5] або мережі глибокого навчання входи комбінуються у другому рівні перед тим, як виводитись. Там більше ваг, а додані з'єднання дозволяють мережі навчатися та вирішувати більш

складні проблеми. Існує багато видів машинного навчання. Будь-який комп'ютерний алгоритм, який може адаптуватися на основі входів із навколишнього середовища - це система навчання. Ось частковий список:

- Нейронні мережі (глибоке навчання чи інше)
- Підтримка векторних машин
- Адаптивне управління
- Ідентифікація системи
- Ідентифікація параметра (може бути такою ж, як і попередня)
- Адаптивні експертні системи
- Алгоритми управління (пропорційне інтегральне похідне управління зберігає інформацію про постійні входи в своєму інтеграторі)

Деякі системи використовують заздалегідь визначений алгоритм і навчаються, підбираючи параметри алгоритму.

Інші створюють модель повністю з даних. Системи глибокого навчання, як правило, належать до останньої категорії.

Нейронні мережі або нейронні мережі є популярним способом реалізації машинного «інтелекту». Ідея полягає в тому, що вони поводяться як нейрони мозку. У цьому розділі ми дослідимо, як працюють нейронні мережі, починаючи з найбільш фундаментальної ідеї з одним нейроном і прокладаючи шлях до багат шарової нейронної мережі. Нашим прикладом для цього буде маятник. Покажемо, як нейронну мережу можна використовувати для вирішення проблеми прогнозування. Це одне з двох застосувань нейронної мережі, прогнозування та класифікація. Почнемо з простого прикладу класифікації. Давайте спочатку розглянемо один нейрон з двома входами. Це показано на рисунку 1.2.

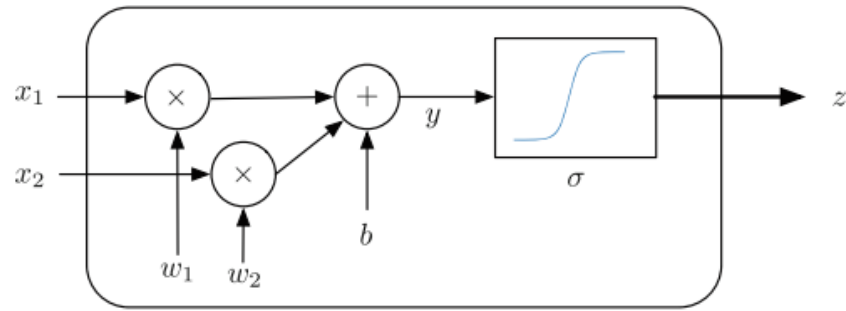


Рисунок 1.2. Нейрон з двома входами

Цей нейрон має входи  $x_1$  і  $x_2$ , зсув  $b$ , ваги  $w_1$  і  $w_2$  і один вихід  $z$ . Функція активації  $\sigma$  приймає зважений вхід і виробляє вихід. На цій діаграмі ми явно додаємо піктограми для кроків множення та додавання в нейроні, але на типових діаграмах нейронних мереж, таких як Рисунок 1.1, вони пропущені.

$$z = \sigma(y) = \sigma(w_1x_1 + w_2x_2 + b), \leftarrow (1.1)$$

Давайте порівняємо це зі справжнім нейроном, як показано на рисунку 1.3.

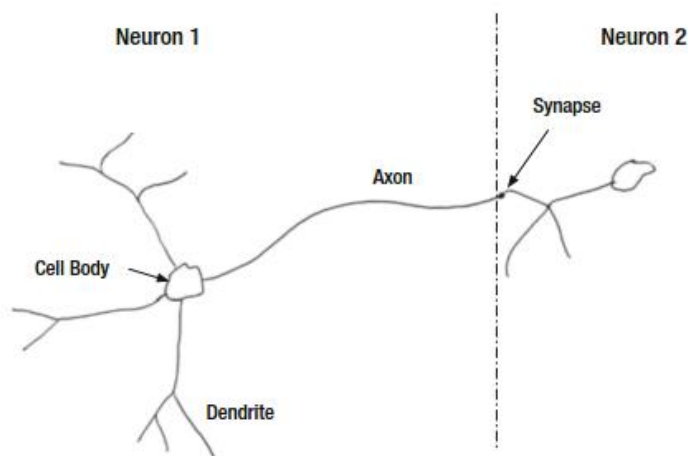


Рисунок 1.3 Справжній нейрон

Справжній нейрон має кілька входів через дендрити. Деякі з них розгалужуються, що означає, що кілька входів можуть підключатися до тіла клітини через один і той же дендрит. Вихід здійснюється через аксон. Кожен нейрон має один вихід. Аксон з'єднується з дендритом через синапс. Сигнали

передаються від аксона до дендриту через синапс. Існує безліч часто використовуваних функцій активації. Покажемо три:

$$\sigma(y) = \tanh(y), \leftarrow (1.2)$$

$$\sigma(y) = \frac{2}{1 + e^{-y}} - 1, \leftarrow (1.3)$$

$$\sigma(y) = y, \leftarrow (1.4)$$

Функції активації, які насичують або досягають значення вхідного сигналу, після чого вихід є постійним або змінюється дуже повільно, моделюють біологічний нейрон, що має максимальну швидкість стрільби. Ці конкретні функції також мають хороші числові властивості, які корисні для навчання. Давайте розглянемо одну вхідну нейронну мережу, зображену на рисунку 1.4. Цей нейрон є

$$z = \sigma(2x + 3), \leftarrow (1.5)$$

де вага  $w$  на одному вході  $x$  дорівнює 2, а зміщення  $b$  дорівнює 3. Якщо функція активації лінійна, нейрон є просто лінійною функцією  $x$ ,

$$z = y = 2x + 3, \leftarrow (1.6)$$

Нейронні мережі використовують лінійні функції активації, часто у вихідному рівні. Саме нелінійні функції активації надають нейромережам їх унікальні можливості.

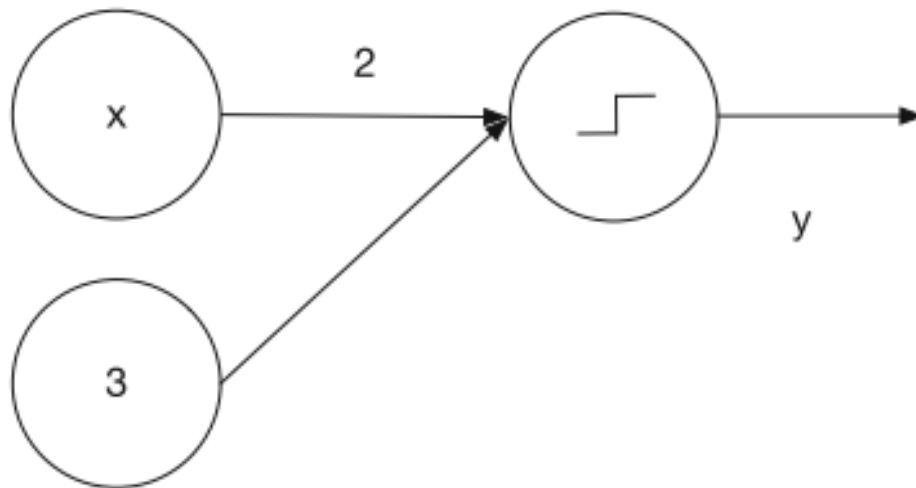


Рисунок 1.4 Нейронна мережа з одним входом

## 1.2 Основні алгоритми та принципи роботи нейронних мереж

Щоразу, коли ми чомусь вчимося, наш мозок зберігає знання. Комп'ютер використовує пам'ять для зберігання інформації. Хоча вони обидва зберігають інформацію, їх механізми дуже різні. Комп'ютер зберігає інформацію у визначених місцях пам'яті, тоді як мозок змінює зв'язок нейронів. Сам нейрон не має можливості зберігання; він просто передає сигнали від одного нейрона до іншого. Мозок - це гігантська мережа цих нейронів, і асоціація нейронів утворює конкретну інформацію. Нейронна мережа імітує механізм мозку. Оскільки мозок складається із зв'язків численних нейронів, нейронна мережа побудована із зв'язками вузлів, які є елементами, що відповідають нейронам мозку. Нейронна мережа імітує асоціацію нейронів, яка є найважливішим механізмом мозку, використовуючи значення ваги.

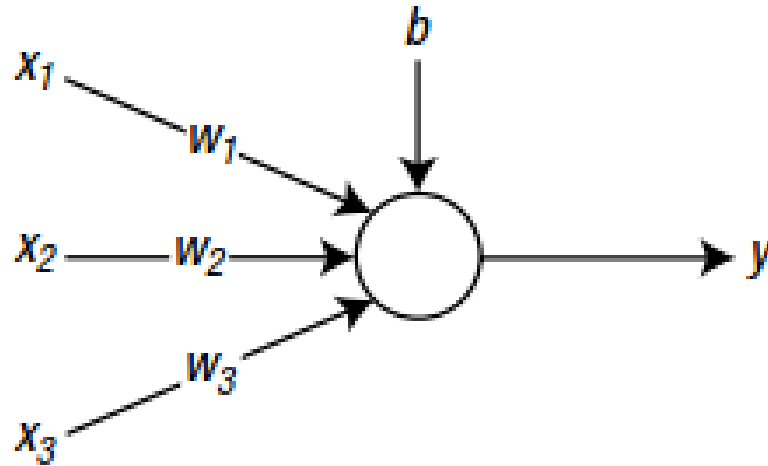


Рисунок 1.5 Вузол, який отримує три входи

Коло і стрілка на малюнку позначають вузол і потік сигналу, відповідно.  $x_1$ ,  $x_2$  та  $x_3$  - це вхідні сигнали.  $w_1$ ,  $w_2$  і  $w_3$  - ваги відповідних сигналів. Нарешті,  $b$  - упередженість, яка є ще одним фактором, пов'язаним із зберіганням інформації. Іншими словами, інформація нейронної мережі зберігається у вигляді ваг та упереджень. Вхідний сигнал із-зовні множиться на вагу, перш ніж він досягне вузла. Як тільки зважені сигнали збираються на вузлі, ці значення додаються до зваженої суми. Зважена сума цього прикладу обчислюється наступним чином:

$$v = (w_1 \times x_1) + (w_2 \times x_2) + (w_3 \times x_3) + b, \leftarrow (1.7)$$

Це рівняння вказує на те, що сигнал із більшою вагою має більший ефект. Наприклад, якщо вага  $w_1$  дорівнює 1, а  $w_2$  дорівнює 5, тоді сигнал  $x_2$  має в п'ятеро більший ефект, ніж ефект  $x_1$ . Коли  $w_1$  дорівнює нулю,  $x_1$  взагалі не передається на вузол. Це означає, що  $x_1$  відключений від вузла. Цей приклад показує, що ваги нейронної мережі імітують, як мозок змінює асоціацію нейронів.

Оскільки мозок є гігантською мережею нейронів, нейронна мережа - це мережа вузлів. Залежно від того, як вузли з'єднані, можуть бути створені різноманітні нейронні мережі. Один з найбільш часто використовуваних типів



нейронних мереж використовує шарувату структуру вузлів, як показано на рисунку 1.6.

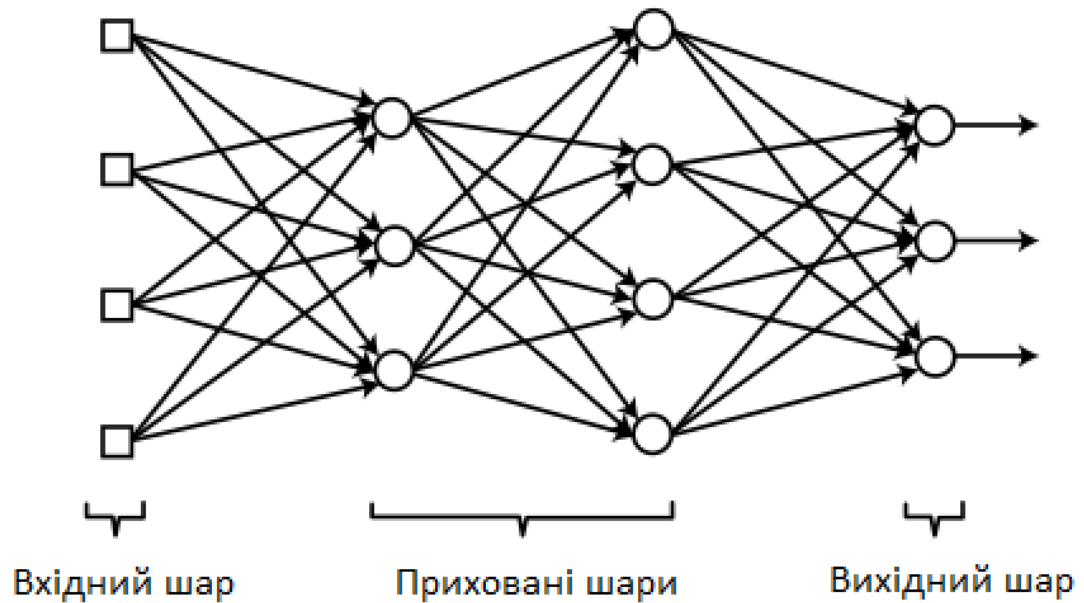


Рисунок 1.6 Шарувата структура вузлів

Група квадратних вузлів на рисунку 1.6 називається вхідним шаром. Вузли вхідного рівня просто діють як прохід, який передає вхідні сигнали наступним вузлам. Тому вони не обчислюють зважену суму та функцію активації. Це є причиною того, що вони позначаються квадратами та відрізняються від інших кругових вузлів. На відміну від цього, група крайніх правих вузлів називається вихідним рівнем. Вихід з цих вузлів стає кінцевим результатом нейронної мережі. Шари між вхідним і вихідним шарами називаються прихованими шарами. Їм дано цю назву, оскільки вони недоступні ззовні нейронної мережі. Нейронна мережа була розроблена від простої архітектури до все більш складної структури. Спочатку піонери нейронних мереж мали дуже просту архітектуру лише з вхідними та вихідними рівнями, які називаються одношаровими нейронними мережами. Коли приховані шари додаються до одношарової нейронної мережі, це створює багатшарову нейронну мережу. Отже, багаторівнева нейронна мережа

складається з вхідного рівня, прихованого рівня (рівнів) і вихідного рівня. Нейромережа, яка має єдиний прихований шар, називається неглибокою нейронною мережею або нейромережею ванілі. Багат шарова нейронна мережа, що містить два або більше прихованих шарів, називається глибокою нейронною мережею. Більшість сучасних нейронних мереж, що використовуються в практичному застосуванні, є глибокими нейронними мережами.

Коротко кажучи, глибоке навчання - це техніка машинного навчання, яка використовує глибоку нейронну мережу. Як відомо, глибока нейронна мережа - це багат шарова нейронна мережа, яка містить два або більше прихованих шарів. Хоча це може бути невтішно просто, це справжня суть глибокого навчання.

Тепер, знаючи, що глибоке навчання - це просто використання більш глибоких (прихованих шарів) нейронних мереж, ви можете запитати: «Що робить глибоке навчання таким привабливим? Хто-небудь коли-небудь думав зробити шари нейронної мережі ще глибшими? " Для того, щоб відповісти на ці питання, нам потрібно заглянути в історію нейронної мережі. Не пройшло дуже багато часу, щоб одношарова нейронна мережа, перше покоління нейронної мережі, виявила свої основні обмеження при вирішенні практичних проблем, з якими стикалося машинне навчання.<sup>1</sup> Дослідники вже знали, що багат шарова нейронна мережа бути наступним проривом. Однак пройшло приблизно 30 років, поки до одношарової нейронної мережі не було додано ще один шар. Можливо, буде не просто зрозуміти, чому так довго тривав лише один додатковий шар. Це було тому, що належного правила навчання для багат шарової нейронної мережі не було знайдено. Оскільки навчання є єдиним способом для нейронної мережі зберігати інформацію, не підготовлена нейромережа марна. Проблема навчання багат шарової нейронної мережі була остаточно вирішена в 1986 році, коли був введений алгоритм зворотного розповсюдження. Нейронна мережа знову була на сцені. Однак незабаром це зіткнулося з іншою проблемою. Його виконання щодо практичних проблем не відповідало очікуванням. Звичайно, були різні спроби подолати обмеження, включаючи додавання прихованих шарів та додавання

вузлів у прихований шар. Однак жоден з них не працював. Багато з них дали ще гірші результати. Оскільки нейронна мережа має дуже просту архітектуру та концепцію, нічого особливого зробити не вдалося б, щоб покращити її. Нарешті, нейронна мережа була засуджена до відсутності можливості вдосконалення, і про неї забули. Він залишався забутим приблизно 20 років до середини 2000-х років, коли було запроваджено Поглиблене навчання, відкривши нові двері. Минув деякий час, щоб глибоко прихований шар дав достатню продуктивність через труднощі в тренуванні глибокої нейронної мережі. У будь-якому випадку, сучасні технології глибокого навчання дають вражаючий рівень продуктивності, що перевершує інші методи машинного навчання, а також інші нейронні мережі, і переважає у дослідженнях штучного інтелекту.

Підводячи підсумок, причиною того, що багатошаровій нейронній мережі знадобилося 30 років, щоб вирішити проблеми одношарової нейронної мережі, була відсутність правила навчання, яке врешті-решт було вирішено алгоритмом зворотного розповсюдження. Навпаки, причиною того, минуло ще 20 років до впровадження глибокого глибокого навчання на основі нейронних мереж - низька продуктивність. Тренування зворотного розповсюдження з додатковими прихованими шарами часто призводило до погіршення продуктивності. Глибоке навчання забезпечило вирішення цієї проблеми.

Потрібно зазначити загальноприйнятну класифікацію нейронних мереж: бінарна та мультикласова. Почнемо з бінарної класифікації нейронної мережі, яка класифікує вхідні дані в одну з двох груп. Цей класифікатор насправді корисний для більшої кількості програм, ніж ви можете очікувати. Деякі типові програми включають фільтрування спаму (спам або звичайна пошта) та схвалення позик (схвалення чи відмова). Для двійкової класифікації для нейронної мережі достатньо одного вихідного вузла. Це пояснюється тим, що вхідні дані можна класифікувати за вихідним значенням, яке або більше, або менше, ніж порогове значення. Наприклад, якщо сигмоїдна функція використовується як функція активації вихідного вузла, дані можна класифікувати за тим, чи є вихід більше 0,5.

Оскільки сигмовидна функція коливається від 0-1, ми можемо розділити групи посередині.

Для даних координат  $(x, y)$  модель визначає, до якої групи належать дані. У цьому випадку дані навчання подаються у форматі, наведеному на рисунку 1.7. Перші дві цифри позначають координати  $x$  та  $y$  відповідно, а символ представляє групу, до якої належать дані. Дані складаються з вхідних даних і правильних результатів, оскільки вони використовуються для контрольованого навчання

{5, 7, $\Delta$ }
{9, 8, $\bullet$ }
...
{6, 5, $\bullet$ }

Рисунок 1.7 Навчальні дані двійкової класифікації

Тепер давайте побудуємо нейронну мережу. Кількість вхідних вузлів дорівнює кількості вхідних параметрів. Оскільки вхідні дані цього прикладу складаються з двох параметрів, мережа використовує два вхідні вузли. Нам потрібен один вихідний вузол, оскільки це реалізує класифікацію двох груп за попередньою адресою. Сигмоїдна функція використовується як функція активації, а прихований шар має чотири вузли. На рисунку 1.8 показана описана нейронна мережа.

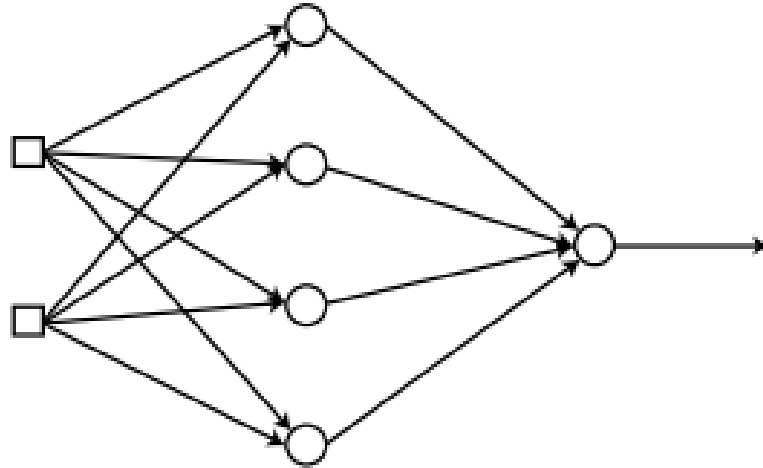


Рисунок 1.8 Нейронна мережа для тренувальних даних

Коли ми тренуємося в цій мережі з даними навчальних даних, ми можемо отримати бінарну класифікацію, яку ми хочемо. Однак є проблема. Нейронна мережа видає числові виходи в діапазоні від 0-1, тоді як ми маємо символічно правильні виходи, що позначаються як  $\Delta$  і  $\bullet$ . Ми не можемо обчислити помилку таким чином; нам потрібно перемкнути символи на цифрові коди. Ми можемо призначити максимальне та мінімальне значення сигмовидної функції для двох класів наступним чином:

Class  $\Delta$   $\rightarrow$  1

Class  $\bullet$   $\rightarrow$  0

Рисунок 1.9 Схема призначення мінімального та максимального значення для класів

Зміна символів класу дає дані про навчання, показані на рисунку 1.10.

{5, 7, 1}
{9, 8, 0}
...
{6, 5, 0}

Рисунок 1.10

Дані навчання, показані на рисунку 1.10, - це те, що ми використовуємо для тренування нейронної мережі. Нейронна мережа двійкової класифікації зазвичай приймає функцію перехресної ентропії попереднього рівняння для навчання. Не обов'язково робити це постійно, але це вигідно для продуктивності та процесу впровадження. Процес навчання нейронної мережі двійкової класифікації [5] узагальнено на наступних етапах. Звичайно, ми використовуємо функцію перехресної ентропії як функцію витрат, а сигмоїдну функцію як функцію активації прихованих і вихідних вузлів.

1) Нейронна мережа двійкової класифікації має один вузол для вихідного рівня. Сигмоїдна функція використовується для функція активації.

2) Треба переключити назви класів даних навчання на цифри використовуючи максимальне та мінімальне значення сигмоїдної функції.

3) Ініціалізувати ваги нейронної мережі адекватними значеннями.

4) Ввести вхідні дані з навчальних даних {input, correct output} у нейронну мережу та отримайте вихідні дані. Обчислити похибку між вихідним та правильним вихідним кодом та визначити дельту,  $\delta$ , вихідних вузлів.

5) Поширити вихідну дельту назад і обчислити дельту наступних прихованих вузлів.

6) Повторювати крок 5, доки не буде можливості дістатись до прихованого шару безпосередньо праворуч від вхідного шару.

7) Відрегулювати ваги нейронної мережі, використовуючи це правило навчання

$$\Delta w_{ij} = \alpha \delta_i x_j, \leftarrow (1.8)$$

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}, \leftarrow (1.9)$$

8) Повторити кроки 4-7 для всіх точок даних тренувань

9) Повторити кроки 4-8, доки нейронна мережа не буде навчена належним чином.

Незважаючи на те, що процес виглядає складним через багато етапів, він в основному такий самий, як і процес зворотного розповсюдження.

Мультикласова класифікація – така, де використовується нейронна мережа, в якій більше як три класи.

Спочатку нам потрібно побудувати нейронну мережу. Ми використовуватимемо два вузли для вхідного рівня, оскільки вхід складається з двох параметрів. Для простоти приховані шари наразі не розглядаються. Нам також потрібно визначити кількість вихідних вузлів. Широко відомо, що узгодження кількості вихідних вузлів з кількістю класів є найбільш перспективним методом. У цьому прикладі ми використовуємо три вихідні вузли, оскільки проблема вимагає трьох класів. Після того, як нейронна мережа пройшла навчання з поданими даними, ми отримуємо багатокласовий класифікатор, який ми хочемо. Дані про навчання наведені на рисунку 1.11. Для кожної точки даних перші два числа - це координати x та y відповідно, а третє значення - відповідний клас. Дані включають вхідні та правильний результат, оскільки він використовується для навчання під контролем.

{ 5, 7, Class1 }
{ 9, 8, Class 3 }
_____
{ 2, 4, Class 2 }
_____
_____
...
_____
{ 6, 5, Class 3 }
_____

Рисунок 1.11. Навчальні дані з багатокласовим класифікатором

Для обчислення помилки ми перетворюємо імена класів на числові коди, як це було зроблено в попередній частині. Враховуючи, що у нас є три вихідних вузли з нейронної мережі, ми створюємо класи як аналогічні вектори. Це перетворення передбачає, що кожен вихідний вузол зіставляється з елементом вектора класу, який дає лише 1 для відповідного вузла. Наприклад, якщо дані належать до класу 2, вихідні дані дають лише 1 для другого вузла і 0 для інших, як показано на рисунку 1.12



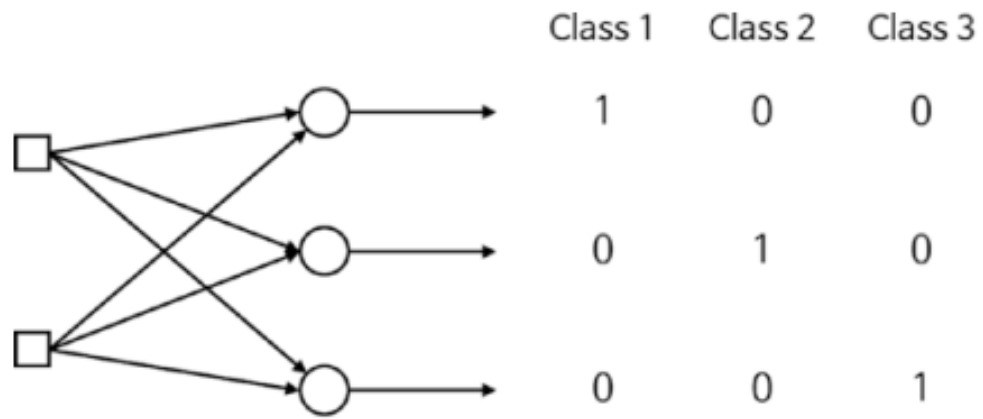


Рисунок 1.12 Кожен вихідний вузол тепер зіставляється з елементом вектора класу

Цей прийом вираження називається одноразовим кодуванням або кодуванням 1 з N. Причиною того, що ми узгоджуємо кількість вихідних вузлів з кількістю класів, є застосування цієї техніки кодування.

Далі слід визначити функцію активації вихідного вузла. Оскільки правильні результати трансформованих навчальних даних коливаються від нуля до одиниці, чи можемо ми просто використовувати сигмоподібну функцію, як це було зроблено для двійкової класифікації? Загалом, багатокласові класифікатори використовують функцію softmax як функцію активації вихідного вузла. Функції активації, про які ми вже говорили, включаючи сигмоподібну функцію, враховують лише зважену суму входів. Вони не враховують результати з інших вихідних вузлів. Однак функція softmax враховує не лише зважену суму входів, а й входи для інших вихідних вузлів. Наприклад, коли зважена сума входів для трьох вихідних вузлів дорівнює 2, 1 та 0,1 відповідно, функція softmax обчислює результати. Усі зважені суми входних даних потрібні в знаменнику.

Здається, перший вузол виведення знаходиться в класі 1 із 100-відсотковою ймовірністю. Тоді дані належать до класу 1? Не так швидко. Інші вихідні вузли також вказують на 100-відсоткову ймовірність потрапляння до класу 2 та класу 3. Отже, адекватна інтерпретація результатів роботи з нейтральної мережі багатокласової класифікації вимагає врахування відносних величин усіх виходів

вузлів. У цьому прикладі фактична ймовірність бути кожним класом дорівнює частці чисел 1 та 3. Функція softmax 3 надає правильні значення. Функція softmax підтримує суму вихідних значень як одну, а також обмежує окремі результати в межах значень 0- 1. Оскільки вона враховує відносні величини всіх виходів, функція softmax є підходящим вибором для нейтральних мереж класифікації багатокласів. Вихід з і-го вихідного вузла функції softmax обчислюється наступним чином:

$$y_i = \varphi(v_i) = \frac{e^{v_i}}{e^{v_1} + e^{v_2} + e^{v_3} + \dots + e^{v_M}}$$

$$= \frac{e^{v_i}}{\sum_{k=1}^M e^{v_k}}, \leftarrow (1.10)$$

де,  $v_i$  - зважена сума і-го вихідного вузла, а  $M$  - кількість вихідних вузлів. Дотримуючись цього визначення, функція softmax задовольняє наступну умову:

$$\varphi(v_1) + \varphi(v_2) + \varphi(v_3) + \dots + \varphi(v_M) = 1, \leftarrow (1.11)$$

Нарешті, слід визначити правило навчання. У багатокласовій нейромережевій класифікаційній мережі зазвичай використовуються перехресні ентропійні правила навчання, як це робить двійкова класифікаційна мережа. Це пов'язано з високою продуктивністю навчання та простотою, яку забезпечує функція перехресної ентропії. Коротше кажучи, правило навчання нейронної мережі багатокласової класифікації ідентичне правилу нейронної мережі двійкової класифікації у попередньому розділі. Хоча ці дві нейронні мережі використовують різні функції активації - сигмоїд для двійкового і м'який макс для мультикласу - виведення правила навчання призводить до того самого результату. Краще мати менше інформації для запам'ятовування.

## 2 ОБҐРУНТУВАННЯ ВИБОРУ НАПРЯМКУ ДОСЛІДЖЕНЬ

### 2.1 Вибір технологій розробки

Оскільки будуть розроблятися спеціальні методи глибокого навчання нейронних мереж, то найбільш підходящим засобом розробки можна відмітити пакет прикладних програм MathWorks та MATLAB разом з мовою програмування, що використана у даному програмному забезпеченні.

MathWorks представляє кілька пакетів для машинного навчання. Їх набори інструментів працюють безпосередньо з MATLAB та Simulink. Продукти MathWorks надають високоякісні алгоритми для аналізу даних, а також графічні інструменти для візуалізації даних. Інструменти візуалізації є критично важливою частиною будь-якої системи машинного навчання. Вони можуть використовуватися для збору даних, наприклад, для розпізнавання зображень або як частина систем для автономного управління транспортними засобами, або для діагностики та налагодження під час розробки. Всі ці пакети можуть бути інтегровані між собою та з іншими функціями MATLAB для створення потужних систем машинного навчання. У моїй роботі я використовую лише Deep Learning ToolBox та всі пов'язані пакети.

Набір інструментів глибокого навчання (Deep Learning ToolBox) дозволяє проектувати, будувати та візуалізувати згорткові нейронні мережі. Ви можете реалізувати наявні перекваліфіковані нейромережі, доступні в Інтернеті, такі як GoogLeNet, VGG-16, VGG 19, AlexNet та ResNet-59. GoogLeNet та AlexNet - це мережі класифікації зображень. Набір інструментів глибокого навчання має широкі можливості для візуалізації та налагодження нейронних мереж. Інструменти налагодження важливі для забезпечення належної поведінки вашої системи та розуміння того, що відбувається у вашій нейронній мережі. Він включає ряд попередньо навчених моделей. Ми будемо використовувати цей набір інструментів у всіх наших прикладах.

Набір інструментів управління приладами MATLAB призначений для безпосереднього підключення інструментів. Це спрощує використання MATLAB з апаратним забезпеченням. Прикладами є осцилографи, генератори функцій та джерела живлення. Набір інструментів забезпечує підтримку TCP / IP, UDP, I2C, SPI та Bluetooth. За допомогою Instrument Control Toolbox можна інтегрувати MATLAB безпосередньо у свій лабораторний робочий процес без необхідності писати драйвери або створювати спеціалізовані файли MEX.

Набір інструментів статистики та машинного навчання надає методи аналізу даних для збору тенденцій та закономірностей із величезного обсягу даних. Ці методи не потребують моделі для аналізу даних. Функції набору інструментів можна розділити на інструменти класифікації, інструменти регресії та інструменти кластеризації. Статистика є основою для глибокого навчання. Методи класифікації використовуються для розміщення даних у різних категоріях. Наприклад, дані у вигляді зображення можуть бути використані для класифікації зображення органу як такого, що має пухлину. Класифікація використовується для розпізнавання почерку, зарахування кредитів та ідентифікації обличчя. Методи класифікації включають машини векторної підтримки (SVM), дерева рішень та нейронні мережі. Методи регресії дозволяють будувати моделі з поточних даних для прогнозування майбутніх даних. Потім моделі можна оновлювати, коли нові дані стануть доступними. Якщо дані використовуються лише один раз для створення моделі, то це пакетний метод. Метод регресії, який включає дані у міру їх надходження, є рекурсивним методом. Кластеризація знаходить природні групування в даних. Розпізнавання об'єктів - це застосування методів кластеризації. Наприклад, якщо ви хочете знайти автомобіль на зображенні, ви шукаєте дані, пов'язані з тією частиною зображення, яка є автомобілем. Хоча автомобілі мають різну форму та розміри, вони мають багато спільних рис. Кластеризація також може мати справу з різними орієнтаціями та масштабуванням. Набір інструментів має безліч функцій для підтримки цих областей, і багато з них, які не входять до цих категорій.

Статистика та набір інструментів машинного навчання надають професійні інструменти, які легко інтегруються в середовище MATLAB.

## 2.2 Обґрунтування вибору середовища розробки програмної системи

Глибоке навчання використовує нейронні мережі для вивчення корисних зображень функцій безпосередньо з даних. Нейронні мережі поєднують в собі кілька нелінійних процесорних шарів, використовуючи прості елементи, що працюють паралельно і натхненні біологічною нервовою системою. Моделі глибокого навчання можуть досягти сучасної точності класифікації об'єктів, іноді перевищуючи показники на рівні людини.

Можна тренувати моделі, використовуючи великий набір маркованих даних та архітектури нейронних мереж, які містять багато шарів, як правило, включаючи деякі згорткові шари. Навчання цим моделям є обчислювально обчислювальним, і зазвичай можна прискорити навчання, використовуючи високопродуктивний графічний процесор.

Багато програм глибокого навчання використовують файли зображень, а іноді і мільйони файлів зображень. Для ефективного доступу до багатьох файлів зображень для глибокого навчання MATLAB надає функцію `imageDatastore`. Цю функцію можна використовувати для:

- автоматичного читання партії зображень для швидшої обробки в програмах машинного навчання та комп'ютерного зору
- імпортування даних із колекцій зображень, які завеликі, щоб поміститися в пам'ять
- позначання даних зображень автоматично на основі імен папок

Трансферне навчання зазвичай використовується в програмах глибокого навчання. Можна взяти заздалегідь навчену мережу і використовувати її як вихідну точку для вивчення нового завдання. Точне налаштування мережі за допомогою навчання на трансфертах набагато швидше і простіше, ніж навчання з нуля. Є можливість швидко змусити мережу вивчити нове завдання, використовуючи меншу кількість навчальних зображень. Перевага трансферного навчання полягає в тому, що попередньо навчена мережа вже вивчила багатий набір функцій, які можна застосувати до широкого кола інших подібних завдань.

Наприклад, якщо взяти мережу, навчену тисячам або мільйонам зображень, ви можете перекваліфікувати її для виявлення нових об'єктів, використовуючи лише сотні зображень. В такому разі необхідно ефективно налаштувати попередньо навчену мережу з набагато меншими наборами даних, ніж вихідні навчальні дані. Якщо в наявності є дуже великий набір даних, то навчання передачі може бути не швидшим, ніж навчання нової мережі.

Трансферне навчання дозволяє:

- перенести вивчені функції попередньо навченої мережі на нову проблему
- трансферне навчання швидше і простіше, ніж навчання новій мережі
- скоротити час навчання та розмір набору даних
- виконання глибокого навчання без необхідності вчити створювати абсолютно нову мережу

Витяг функцій дозволяє використовувати потужність попередньо навчених мереж, не витрачаючи часу та зусиль на навчання. Вилучення функцій може бути найшвидшим способом використання глибокого навчання. Можна витягнути вивчені функції із заздалегідь навченої мережі та використати ці функції для підготовки класифікатора, наприклад, машини з підтримкою векторів (SVM - потрібна статистика та інструментарій машинного навчання). Наприклад, якщо SVM, який навчається за допомогою alexnet, може досягти > 90% точності у вашому навчальному та перевірочному наборах, то тонка настройка з навчанням

передачі може не коштувати зусиль для отримання додаткової точності. Якщо виконувати точну настройку на невеликому наборі даних, є ризик перенасичення. Якщо SVM не може досягти достатньо точної для додатка, то тонка настройка коштує зусиль для досягнення більш високої точності.

Можна взяти попередньо навчену мережу класифікації зображень, яка вже навчилася витягувати потужні та інформативні особливості із природних зображень та використовувати її як вихідну точку для вивчення нового завдання. Більшість попередньо навчених мереж навчаються на підмножині бази даних ImageNet, яка використовується у широкомасштабному завданні візуального розпізнавання ImageNet (ILSVRC). Ці мережі пройшли навчання понад мільйона зображень і можуть класифікувати зображення за 1000 категоріями об'єктів, таких як клавіатура, кружка для кави, олівець та багато тварин. Використання попередньо навченої мережі з навчанням передачі, як правило, набагато швидше і простіше, ніж навчання мережі з нуля.

Попередньо оброблені мережі мають різні характеристики, які важливі при виборі мережі, яка застосовується до вашої проблеми. Найважливішими характеристиками є точність мережі, швидкість та розмір. Вибір мережі, як правило, є компромісом між цими характеристиками. Використовуйте графік нижче, щоб порівняти точність перевірки ImageNet з часом, необхідним для прогнозування за допомогою мережі.

Хороша мережа має високу точність і швидкість роботи. Графік відображає точність класифікації в порівнянні з часом прогнозування при використанні сучасного графічного процесора та розміром міні-партії 128. Час передбачення вимірюється щодо найшвидшої мережі. Площа кожного маркера пропорційна розміру мережі на диску.

Точність класифікації в наборі перевірки ImageNet є найпоширенішим способом вимірювання точності мереж, які навчаються на ImageNet. Мережі, які є точними в ImageNet, часто також точні, коли ви застосовуєте їх до інших наборів даних природних зображень, використовуючи навчання передачі або вилучення

функцій. Це узагальнення можливо, оскільки мережі навчилися витягувати потужні та інформативні особливості із природних зображень, які узагальнюються на інші подібні набори даних. Однак висока точність ImageNet не завжди передається безпосередньо до інших завдань, тому бажано спробувати кілька мереж.



### 3 РОЗРОБКА СКЛАДОВИХ ПРОГРАМНОГО КОМПЛЕКСУ

#### 3.1 Моделювання літального апарату

До широкої доступності GPS, Loran та інших електронних навігаційних засобів пілоти використовували візуальні сигнали з місцевості для навігації. Зараз усі користуються GPS. Згідно контексту поставленого завдання будуть розроблені методи, які зможуть співвідносити місцевості бази даних. Потім вони використовуватимуть цю інформацію, щоб визначити напрями польоту об'єкта.

Рух точкової маси через тривимірний простір має 3 ступені свободи. Тому нашій літаковій моделі надається 3 ступені просторової свободи. Вектор швидкості виражається як відносна за вітром величина ( $V$ ) із напрямними компонентами для напрямку ( $\psi$ ) та кута траєкторії польоту ( $\gamma$ ). Положення є прямим інтегралом швидкості і виражається в  $y$  = північ,  $x$  = схід,  $h$  = вертикальні координати. Крім того, тяга двигуна змодельована як система першого порядку, де постійну часу можна змінити, щоб наблизити час відгуку двигуна різних літаків.

На рисунку 3.1 зображена схема вектора швидкості в системі координат Північ-Схід-Вгору. Похідні часу беруться в цих рамках. Це не суто інерційна система координат, оскільки вона обертається із Землею. Однак швидкість обертання Землі досить мала в порівнянні зі швидкістю повороту літака, щоб нею можна було безпечно нехтувати.

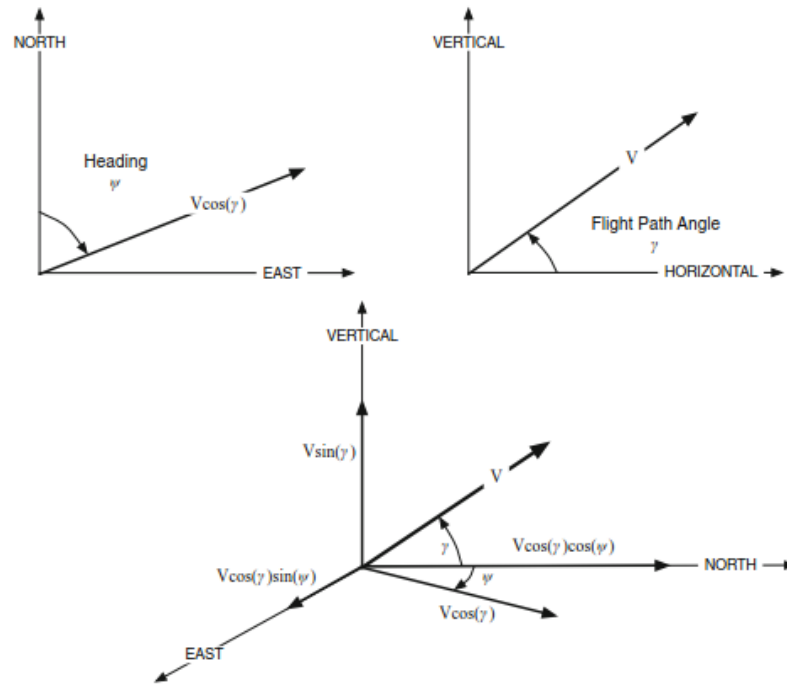


Рисунок 3.1 Схема вектора швидкості в системі координат Північ-Схід-Вгору

Точкові маси рівняння руху літака складають

$$\dot{v} = (T \cos \alpha - D - mg \sin \gamma)/m - f_v, \leftarrow (3.1)$$

$$\dot{\gamma} = \frac{1}{mv} ((L + T \sin \alpha) \cos \phi - mg \cos \gamma + f_\gamma), \leftarrow (3.2)$$

$$\dot{\psi} = \frac{1}{mv \cos \gamma} ((L + T \sin \alpha) \sin \phi - f_\psi), \leftarrow (3.3)$$

$$\dot{x}_e = v \cos \gamma \sin \psi + W_x, \leftarrow (3.4)$$

$$\dot{y}_n = v \cos \gamma \cos \psi + W_y, \leftarrow (3.5)$$

$$\dot{h} = v \sin \gamma + W_h, \leftarrow (3.6)$$

$$\dot{m} = -\frac{T}{u_e}, \leftarrow (3.7)$$

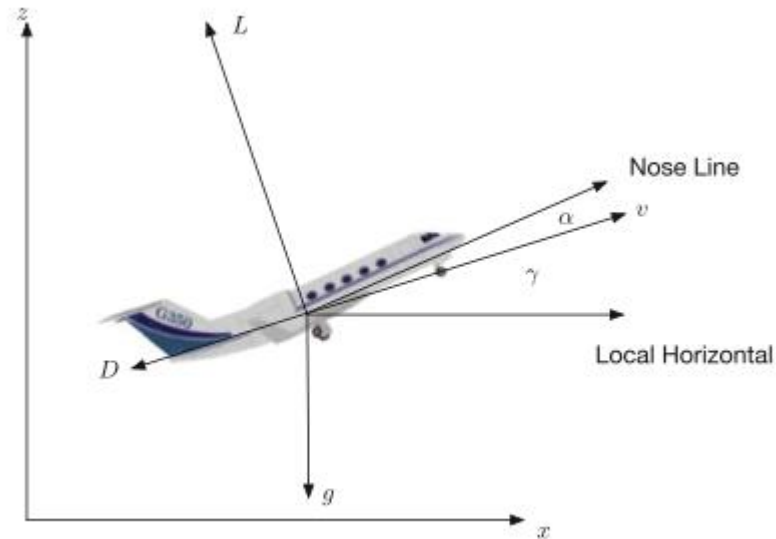


Рисунок 3.2 Модель літака, що показує підйом, тягу та гравітацію

де  $v$  - справжня швидкість,  $T$  - тяга,  $L$  - підйом,  $g$  - прискорення сили тяжіння,  $\gamma$  - відносний кут траєкторії польоту,  $\psi$  - відносний напрямок руху повітря (виміряний за годинниковою стрілкою з півночі),  $\phi$  - кут нахилу берега,  $x$  та  $y$  - східне та північне положення відповідно,  $h$  - висота над рівнем моря. Маса - це загальна суха маса та маса палива. Терміни  $\{f_v, f_\gamma, f_\psi\}$  представляють додаткові сили, зумовлені невизначеністю моделювання, а доданки  $\{W_x, W_y, W_h\}$  є складовими швидкості вітру. Якщо вертикальна швидкість вітру дорівнює нулю, то  $\gamma = 0$  виробляє рівневий політ.  $\alpha$ ,  $\phi$  і  $T$  - елементи управління. На рисунку 1 показані поздовжні символи літака.  $\gamma$  - кут між вектором швидкості та горизонталлю.  $\alpha$  - кут атаки між носом літака і вектором швидкості. Крила можуть бути орієнтовані або мати аеродинамічні профілі, що дають підйом під нульовим кутом атаки. Опір є протилежною величиною до швидкості, а підйом перпендикулярний опору. Підйом повинен врівноважувати силу тяжіння та будь-який компонент перетягування вниз; інакше літак опуститься.

Ми використовуємо дуже просту аеродинамічну модель. Коефіцієнт підйому визначається як

$$C_L = C_{L\alpha} \alpha, \leftarrow (3.8)$$

Коефіцієнт підйому насправді є нелінійною функцією кута атаки. Він має максимальний кут атаки, вище якого крило зупиняється і весь підйом втрачається. Для плоскої пластини  $C_{L\alpha} = 2\pi$ . Коефіцієнт опору становить

$$C_D = C_{D_0} + \frac{C_L^2}{\pi A_R \epsilon}, \leftarrow (3.9)$$

де  $A_R$  - співвідношення сторін, а  $\epsilon$  - коефіцієнт ефективності Освальда, який зазвичай становить від 0,8 до 0,95. Коефіцієнт корисної дії - це те, наскільки ефективно піднімається пара, щоб тягнути. Якщо воно менше одиниці, це означає, що підйомник створює більше опору, спричиненого підйомом, ніж ідеальний. Співвідношення сторін - це співвідношення прольоту крила (від точки, найближчої до фюзеляжу, до кінчика) і хорди (довжина спереду до задньої частини крила).

Динамічний тиск, тиск внаслідок руху літака, становить

$$q = \frac{1}{2} \rho v^2, \leftarrow (3.10)$$

де  $v$  - швидкість, а  $\rho$  - атмосферна щільність. Це тиск на вашу руку, якщо ви висунете її з вікна автомобіля, що рухається. Сили підйому та опору

$$L = q C_{Ls}, \leftarrow (3.11)$$

$$D = q C_{Ds}, \leftarrow (3.12)$$

де  $s$  - змочена площа. Змочена ділянка - це поверхня літального апарата, що виробляє підйом і опору. Ми робимо його однаковим для підйому та опору, але у справжньому літаку деякі частини літака викликають опір (наприклад, ніс), але не роблять жодного підйому. По суті, ми припускаємо, що літак - все крило. Ми

створюємо функцію правої сторони для моделі. Це буде викликано функцією числового інтегрування. Динамічну модель наведено в лістингу 1 додатку В.

Структура даних за замовчуванням визначена у підфункції `DefaultDataStructure`. Структура даних включає як постійні параметри, так і входи управління.

### Лістинг 3.1 Постійні параметри і входи управління

```
d=struct('cD0',0.01,'aR',2.67,'eps',0.95,'cLAlpha',2*pi,'s',64.52,...
'g',9.806,'alpha',0,'phi',0,'thrust',0,'m',19368.00,...
'f',zeros(3,1),'W',zeros(3,1));
```

Ми використовуємо модифіковану експоненціальну атмосферу для щільності:

### Лістинг 3.2 Модифікована експоненціальна атмосфера для щільності

```
function rho = Density( h )
rho = 1.225*exp(-0.0817*h^1.15);
```

Ми хочемо підтримувати баланс сил, щоб швидкість літака була постійною, а літак не міняв кут траєкторії польоту. Наприклад, при рівному польоті літак не піднімався і не спускався. Нам потрібно керувати літаком у рівному польоті, щоб швидкість залишалася постійною і  $\gamma = 0$  для будь-якого  $\phi$ . Відповідними рівняннями є

$$0 = T \cos \alpha - D, \leftarrow (3.13)$$

$$0 = (L + T \sin \alpha) \cos \phi - mg, \leftarrow (3.14)$$

Нам потрібно знайти  $T$  і  $\alpha$  з урахуванням  $\phi$ . Простий спосіб - використовувати `fminsearch`. Він викличе `RHSPointMassAircraft` і чисельно знайде елементи керування, які для даного  $\psi$ ,  $h$  та  $v$  мають похідні від часу нуля. Наступний код знаходить рівноважний кут атаки та тяги. RHS викликається `fminsearch`. Код наведено у лістингу 2 додатку В. Він повертає скалярну вартість,

яка є квадратичним значенням прискорення (похідної від часу швидкості) та похідної кута траєкторії польоту. Наше початкове припущення - величина тяги, яка врівноважує опору. Навіть при вгадуванні кута атаки 0 він збігається із набором параметрів за замовчуванням `opt = optimset('fminsearch')`.

Демонстрація для Gulfstream 350, що летить зі швидкістю 250 м/с і висотою 10 км, показана у лістингу 3 додатку В. Результати демонстрації, показані нижче, цілком обґрунтовані. З цими значеннями літак повернеться без зміни висоти або швидкості

### Лістинг 3.3 Результати демонстрації

```
Thrust          7614.63 N
Altitude        10.00 km
Angle of attack  2.41 deg
Bank angle      22.92 deg
Turn radius     15.08 km
```

## 3.2 Створення моделі місцевості

Потрібно створити штучну модель рельєфу з набору рельєфних «плиток». Плитка - це сегмент рельєфу із загальної картини. Ми почнемо зі складання бази даних черепичних ландшафтів. Вони знаходяться в папці місцевості в пакеті MATLAB. Сегмент папки місцевості показаний на рисунку 3.3. Це лише один із способів отримати рельєфну плитку. Існують інтернет-джерела для завантаження плиток. Крім того, у багатьох іграх симуляторів польотів є великі бібліотеки для місцевості. Назва папки - широта і довгота.

Наприклад, -10-10 - це  $-10$  градусів широти та  $-10$  градусів довготи. Наша база даних поширюється лише на  $\pm 60$  градусів широти. Перший блок створює список папок на місцевості.

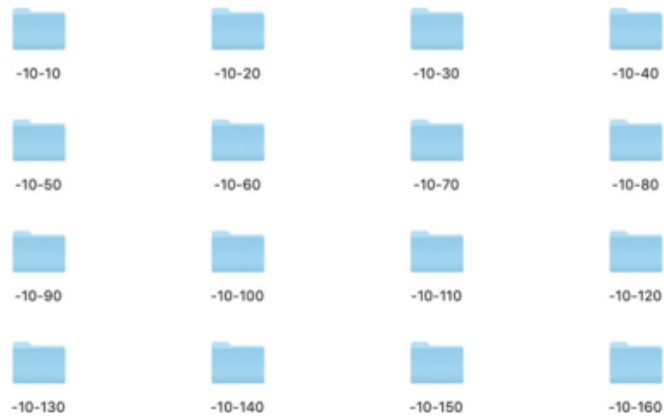


Рисунок 3.3 Сегмент папки з даними про місцевість

Код, наведений в додатку В лістингу 3, знаходить потрібні блоки з даними в папка та створює імена файлів на основі наших широт та довгот. Створюються правильно відформатовані рядки. Це показує один із способів створення рядків. Потрібно звернути увагу, що ми використовуємо % d для створення цілих чисел. Це автоматично робить їх потрібної довжини. Потрібно перевірити наявність позитивних і негативних значень, щоб знаки + і - були правильними.

Далі алгоритм читає зображення, гортає його догори дном і масштабує зображення. Зображення трапляються на північ вниз і південь вгору. Наступний блок коду викликає зображення, щоб намалювати кожне зображення у правильному місці на карті з розміром 3 на 3.

Лістинг 3.1 Зображення частин карти

```
x=0;
y=0;
for k = 1:9
    image('xdata',[x;x+del],'ydata',[y;y+del],'cdata',im{k});
    hold on
    x = x + del;
    if(x == 1X)
        x = 0;
        y = y + del;
    end
end
```

Підфункція `ScaleImage`, показана у Додатку В у лістингу 4 масштабує зображення, виконуючи середнє значення пікселів, зменшених до 1 пікселя. Демонстрація визначає широту та довготу на Близькому Сході. Результати - зображення розміром 3 на 3, зображене на рисунку 3.4. Ми не будемо використовувати це зображення для нейронної мережі, оскільки це буде занадто низька роздільна здатність для будь-чого, крім супутника.



Рисунок 3.4 Розбите зображення місцевості

Нам потрібно збільшити роздільну здатність. Треба підлаштувати код місцевості, щоб створити невеликий сегмент місцевості з більш високою роздільною здатністю, придатну для експериментів з комерційним безпілотником.

Попередній код місцевості добре би працював для орбітального супутника, але не так добре для безпілотника. Згідно з правилами FAA, максимальна висота для малих безпілотних літальних апаратів становить близько 400 футів чи 122 метри. Супутник на низькій орбіті Землі (LEO) зазвичай має висоту 300-500 км. Таким чином, безпілотники, як правило, приблизно в 2500-4000 разів ближче до поверхні, ніж супутник! Спеціалізуємо код для читання лише на чотирьох



зображеннях. Це набагато простіше, ніж CreateTerrain, і менш гнучко. Лістинг коду функції CreateTerrainClose наведено в додатку В у лістингу 5

В результаті ми отримуємо зображення 2 на 2 градуси, його наведено на рисунку 3.5



Рисунок 3.5 Крупний план місцевості

### 3.3 Створення моделі камери

Потрібно створити модель камери для нашої системи глибокого навчання. Знадобиться модель, яка імітує функцію камери, встановленої на безпілотнику. Врешті-решт, використаємо цю модель камери як частину системи навігації на основі місцевості, і ми застосуємо методи глибокого навчання для здійснення навігації по місцевості. Змоделюємо камеру-обскуру та створимо літак на висоті. Пінхол-камера - це наближення найнижчого порядку до реальної оптичної системи. Потім побудуємо симуляцію та продемонструємо камеру.

Раніше вже була створена модель літака. Доповненням буде модель місцевості та модель камери. Пінхол-камера показана на рисунку 3.6. Пінхол-камера має нескінченну глибину різкості, а зображення прямолінійні.

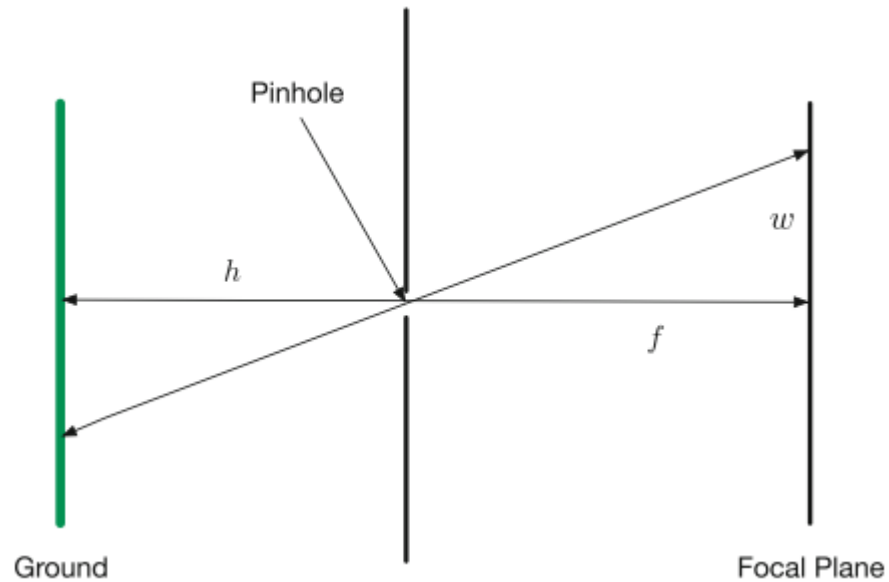


Рисунок 3.6 Камера-обскура.

Точка  $P(x, y, z)$  відображається співвідношеннями на площині зображення

$$v = \frac{fy}{h}, \leftarrow (3.15)$$

$$u = \frac{fx}{h}, \leftarrow (3.16)$$

де  $u$  і  $v$  - координати у фокальній площині,  $f$  - фокусна відстань,  $h$  - відстань від отвору до отвору до точки вздовж осі, нормальної до фокальної площини. Це передбачає, що вісь  $z$  координатного кадру  $x, y, z$  суміщена з оглядом камери. Кут, який бачить чіп зображення, дорівнює

$$\theta = \tan^{-1}\left(\frac{w}{2f}\right), \leftarrow (3.17)$$

де  $f$  - фокусна відстань. Чим коротша фокусна відстань, тим більше зображення. Пінхол-камера не має жодної глибини різкості, але це неважливо для цієї проблеми візуалізації у далекому полі. Поле зору пінхол-камери обмежене лише чутливим елементом. Справжні камери мають лінзи, і зображення не є ідеальними в усьому масиві зображень. Це представляє практичні проблеми, які необхідно вирішити в реальних системах машинного зору.

Фактично не імітується жодної камери. Натомість наша модель камери виробляє карти розміром 16 на 16 пікселів з урахуванням позиції. Результатом є структура даних з координатами  $x$  та  $y$  та зображення. Якщо жодних введених даних не буде введено, це створить плиткову карту зображення.

Тестове зображення масштабоване до розміру 672 на 672 пікселі, також вказане налаштування для кольорів. Код для роботи із зображенням показаний у додатку В у лістингу 6. Ми конвертуємо все зображення у пікселі. На рисунку 3.7 показане вихідне зображення, а потім зображення камери.

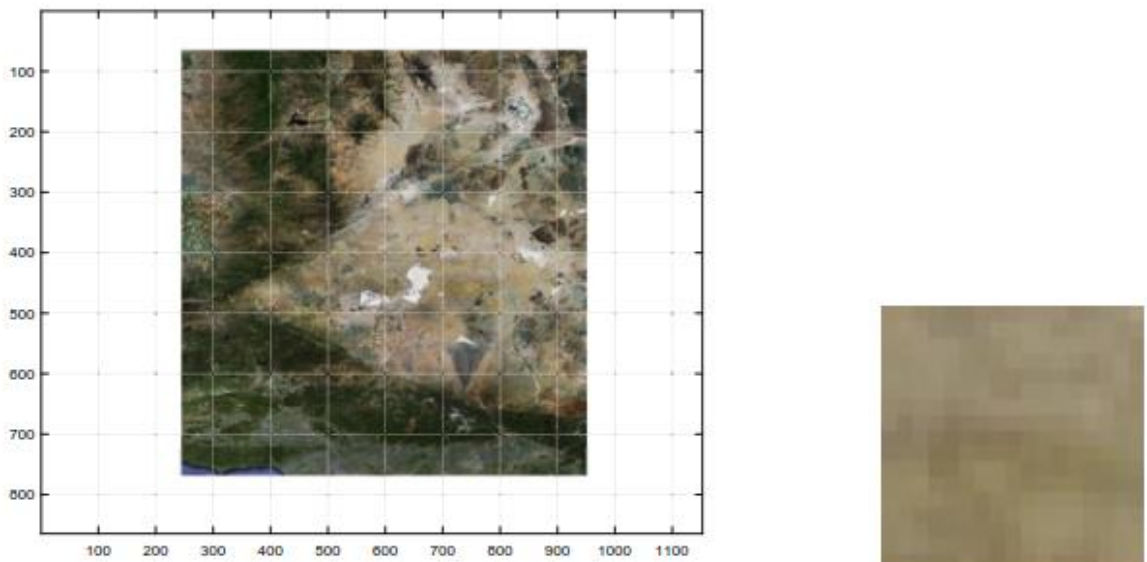


Рисунок 3.7 Вихідне зображення камери місцевості та вигляд камери.

Вигляд камери становить  $16 \times 16$  пікселів.

Зображення місцевості з камери розмите, оскільки на ньому так мало пікселів.

### 3.4 Побудова траєкторії на зображенні

Потрібно побудувати траєкторію поверх зображення. Для цього створимо функцію `PlotXYTrajectory`, яка представлена в додатку В в лістингу 7. Функція `Demo` відображає коло поверх зображення поверхні землі.

Це показано на рисунку 3.8

#### Лістинг 3.4

```
function Demo
i=imread('TerrainClose.jpg');
a=linspace(0,2*pi);
x=[30*cos(a);35*cos(a)];
y=[30*sin(a);35*sin(a)];
PlotXYTrajectory(x,y,i,111,'Trajectory')
```

Хоча система глибокого навчання буде аналізувати всі пікселі на зображенні, цікаво подивитися, як середні значення пікселів для кожного кольору змінюються для кожного зображення на зображенні.

Це показано на малюнку 3.10. Вісь  $x$  - це номер зображення, що проходить по рядках константи  $y$ . Як бачимо, існують значні відмінності навіть у зображеннях поблизу. Це вказує на те, що є достатньо інформації на кожному зображенні для нашої системи глибокого навчання, щоб мати можливість знаходити місця. Це також показує, що можливо було б просто використовувати середні значення для ідентифікації місця. Пам'ятайте, що кожне зображення відрізняється від попереднього лише на 16 пікселів.

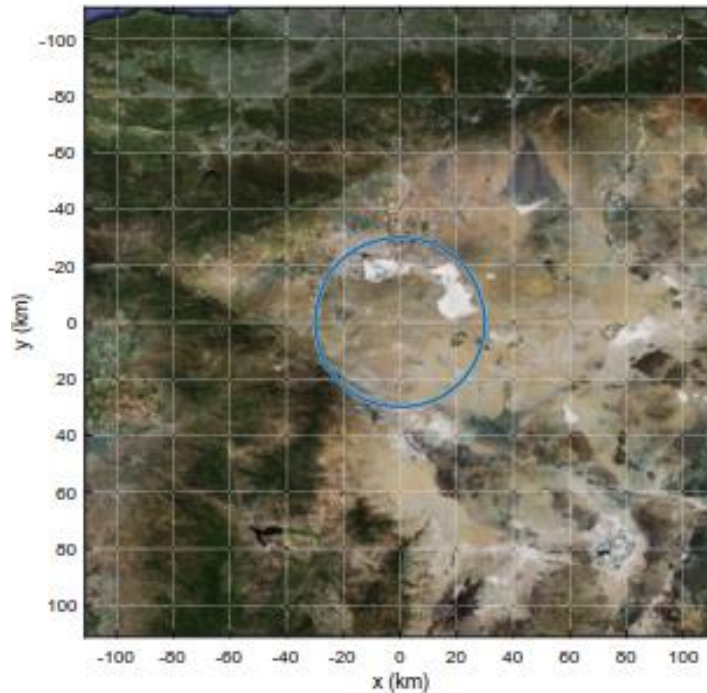


Рисунок 3.8 Траєкторія

### 3.5 Створення тестових зображень

Потрібно створити тестові зображення для моделі поверхні землі. Для цього треба написати скрипт, щоб зчитати 64 на 64 бітне зображення та побудувати навчальні зображення.

Спочатку ми створюємо 64-розрядну версію нашої місцевості, використовуючи будь-яку програму обробки зображень. Це вже було зроблено раніше, тепер буде збережене як `TerrainClose64.jpg`. Наступний сценарій читає зображення та генерує навчальні зображення, зміщуючи індекс по одному пікселю за раз. Зберігаємо зображення в папці `TerrainImages`. Ми також створюємо мітки. Кожне зображення має різну мітку. Для кожного фрагменту місцевості, ми створюємо  $nN$  копій із шумом. Таким чином, буде  $nN$  зображень з міткою 1. Кодом додаємо шум

### Лістинг 3.5 Додавання шуму до зображення

```
uint8 (floor (sig*rand (nBits, nBits, 3)))
```

оскільки шум повинен бути uint8, як зображення. Якщо не перейти на uint8, отримаємо повідомлення про помилку. Також можливий варіант вибрати різні кроки, тобто переміщення зображень більше ніж на 1 піксель. Перший код налаштовує обробку зображень. Обираємо 16-розрядні зображення, оскільки (після наступного етапу навчання) в кожному зображенні достатньо інформації для класифікації кожного. 8-бітне зображення не підходить у даній ситуації.

Також нам треба упевнитись, що серійний номер зображення буде унікальним та упорядкованим. Цей рядок дуже важливий. Це гарантує, що імена відповідають різним зображенням. Виникає потреба робити навчальні копії кожного зображення.

### Лістинг 3.10 Унікальні ідентифікатори

```
kAdd = 10^ceil(log10(nI*nI*nN));
```

Скрипт, що виконує маніпуляції з зображеннями, а саме розрізання зображення на частини, додавання шуму, а також перевірка RGB схеми для кожного з зображень, наведено в додатку В в лістингу 8.

На рисунку 3.9 видно, що зображення дійсно охоплюють територію. Було перевірено, що сума R, G та B була різною для кожного зображення. Це вказує на те, що інформації достатньо для алгоритм машинного навчання.

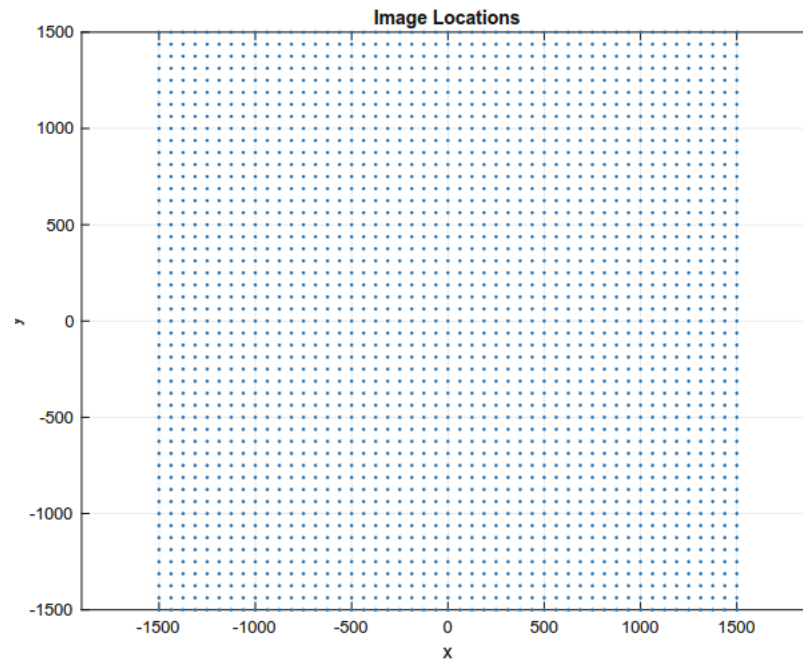


Рисунок 3.9 Показує, що зображення покриває ландшафт

### 3.6 Тренування і тестування

Потрібно створити та протестувати згорткову нейронну мережу. Нейронну мережу навчатимуть асоціювати зображення з розташуванням  $x$  та  $y$ .

Вона буде тренуватися на зображеннях, створених раніше, і зможе повернути координати  $x$  та  $y$ . Світові нейронні мережі широко використовуються для ідентифікації зображень. Повний лістинг методу тренування нейронної мережі наведено у додатку В в лістингу 9.

Маємо шар зображення, який можна прочитати на кожному зображенні. Далі згортаємо їх за допомогою фільтрів [6]. Ваги фільтрів визначаються під час навчання. Ми нормалізуємо виходи і проходимо через функцію активації `relu`. Пул стискає дані. Заповнення встановлює вихідний розмір, рівний вхідному розміру. Як видно з роздруківки шарів, відступ не потрібен, оскільки зображення однакового розміру. Перший шар має вісім фільтрів розміром 3 на 3 пікселі.



Другий шар має 32 фільтри розміром 3 на 3 пікселі. Остаточний набір шарів використовується для класифікації зображень. Як зазначалося в попередньому розділі, кожне зображення має унікальний «клас», який пов'язаний з його місцезнаходженням. Ми використовуємо постійну швидкість навчання. Розмір вхідних даних менший за той, що є за замовчуванням. На рисунку 3.10 показано деякі зображення. На рисунку 3.11 показано вікно навчання. Він здатний класифікувати зображення за 7 епох. Різниця між двома сусідніми зображеннями становить лише 16 пікселів. Даних не так багато, але нейронна мережа може класифікувати кожне зображення зі 100% точністю. У кожному епоху на малюнку 3.11 він обробляє всі навчальні дані.



Рисунок 3.10 Ці вибрані зображення рельєфу показують, що класифікує нейронна мережа.



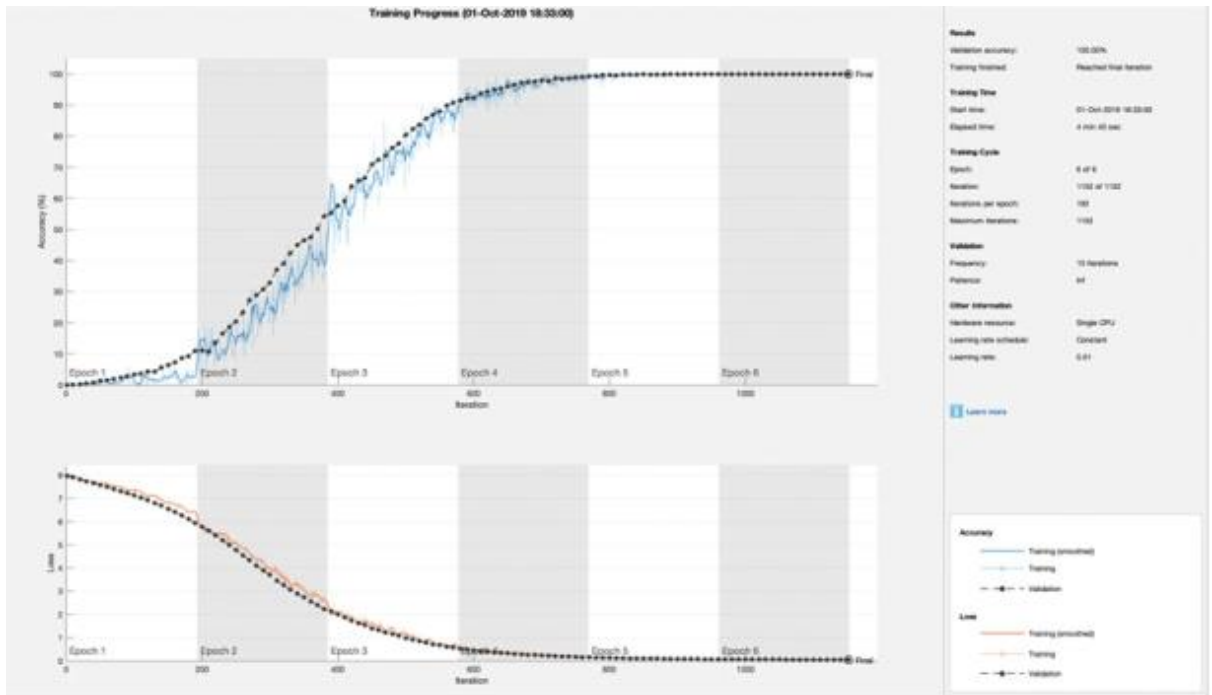


Рисунок 3.11 Вікно тренування

У кожну епоху на рисунку 3.11 алгоритм обробляє всі навчальні дані. На навчальному вікні можна спостерігати такі статистичні дані, як введення зображення, згортка, нормалізація партії, ReLU, максимальне об'єднання, softmax, вихід класифікації та інші. В результаті отримуємо 100% точність. Можна дослідити зміну кількості шарів і спробувати різні функції активації.

### 3.7 Симуляція

Необхідно перевірити наш алгоритм глибокого навчання, використовуючи нашу модель рельєфу. Для цього побудуємо симуляцію, використовуючи треновану нейронну мережу. Потрібно відтворити попередню симуляцію та видалити зайві дані, щоб можна було зосередитись на нейронній мережі та працювати треба з навченою нейронною мережею.

Нейронна мережа класифікує зображення, отримане камерою. Перетворюємо категорію в ціле число, використовуючи `int32`. У підвідображенні відображається зображення, яке нейронна мережа ідентифікує як відповідне зображення камери та зображення камери. Цикл моделювання зупиняється, якщо висота  $x$  (6) менше 1. Скрипт симуляції наведено у додатку В в лістингу 10

На рисунку 3.12 показано траєкторію та вигляд камери. Імітуємо одне повне коло. Нейронна мережа класифікує місцевість, яку вона бачить. Місце кожного зображення зчитується і використовується для побудови траєкторії.

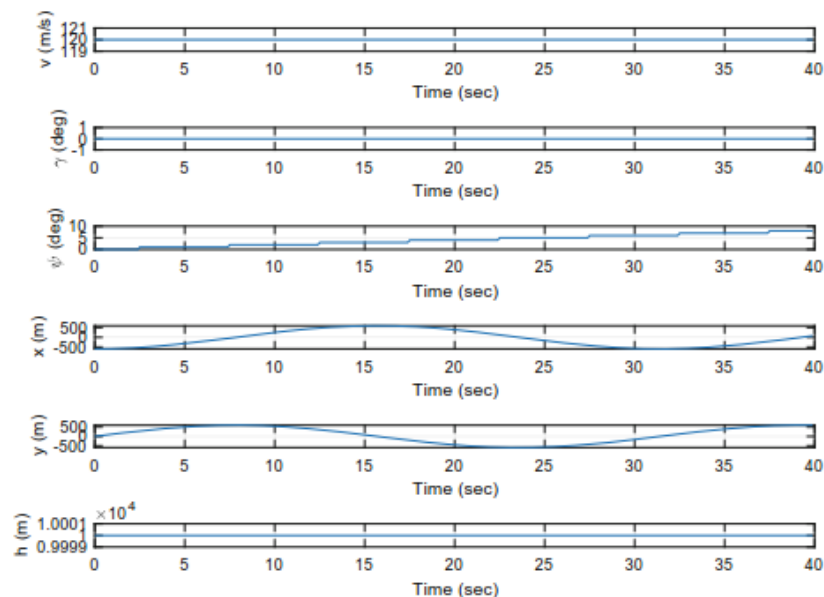


Рисунок 3.12 Вид з камери та траєкторія (одне повне коло)

Двовимірні траєкторія показана на рисунку 3.13 для кругового шляху. Ми переконуємось, що знаходимося в регіонах, де кожне зображення має зміну на один піксель від попереднього зображення. У кутах камера залишалася на одному зображенні, поки це зображення не вийшло. По краях є одна рамка зображення. Якби ми були в цьому регіоні, роздільна здатність була б низькою. Траєкторія від зображень досить близька до фактичної траєкторії. Для кращих результатів потрібна вища роздільна здатність. На практиці виміряні позиції були б вхідними

даними для фільтра Калмана, який моделював динаміку літака, наведений раніше. Це згладить траєкторію та підвищить точність.

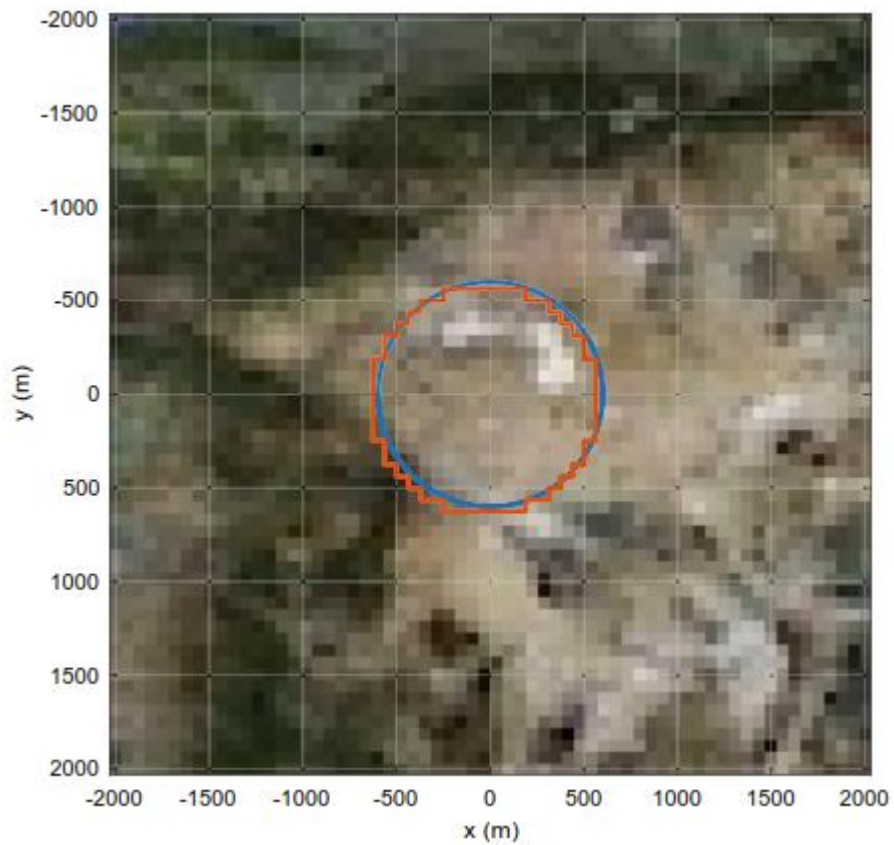


Рисунок 3.13 Визначені сегменти місцевості та траєкторія літака.

## 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 4.1 Охорона праці

Створення програмного забезпечення повинно завжди відбуватись в умовах нанесення мінімальної шкоди здоров'ю розробника. Метою дипломної роботи магістра є розробка методів глибокого машинного навчання для середовища MATLAB у розв'язанні задач навігації, що означає виконання великого обсягу роботи за персональним комп'ютером протягом тривалого часу, тому є доцільним відзначити важливість дотримання норм охорони праці та техніки безпеки під час роботи з електронно-обчислювальними машинами. Найбільш повним нормативним документом щодо забезпечення охорони праці користувачів ПК є “Державні санітарні норми і правила роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин” ДСанПіН 3.3.2.007-98. Даний документ описує: вимоги до виробничих приміщень для експлуатації ВДТ ЕОМ та ПЕОМ, гігієнічні вимоги до параметрів виробничого середовища приміщень, гігієнічні вимоги до організації і обладнання робочих місць, вимоги до режимів праці і відпочинку, вимоги до профілактичних медичних оглядів. Значне зниження наслідків несприятливої дії на програмістів шкідливих та небезпечних факторів можна досягти за допомогою дотримання цих вимог.

Відповідно до встановлених гігієнічно-санітарних вимог (ГОСТ 12.1.005-88, СН 4088-86) роботодавець зобов'язаний забезпечити в приміщеннях з ВДТ оптимальні параметри виробничого середовища.

Природне освітлення в приміщеннях з ВДТ має здійснюватися через вікна, орієнтовані переважно на північ або північний схід і забезпечувати коефіцієнт природної освітленості не нижче ніж 1,5 %. Для захисту від прямих сонячних променів, які створюють прямі та відбиті відблиски з поверхні екранів ПК і клавіатури повинні бути передбачені сонцезахисні пристрої, вікна повинні мати жалюзі або штори.

Основні вимоги до виробничого приміщення для експлуатації ВДТ:

- воно не може бути розміщено у підвалах та цокольних поверхах;
- площа на одне робоче місце в такому приміщенні повинна становити не менше 6,0м<sup>2</sup>, а об'єм не менше 20,0 м<sup>3</sup>;
- воно повинно мати природне та штучне освітлення відповідно до ДБН В.2.5-28:2018;
- в ньому мають бути шафи для зберігання документів, магнітних дисків, полиці, стелажі, тумби тощо, з урахуванням вимог до площі приміщення;
- щоденно проводити вологе прибирання;
- поруч з приміщенням для роботи з ВДТ мають бути обладнані:
- побутова кімната для відпочинку під час роботи;
- кімната психологічного розвантаження.

Штучне освітлення в приміщеннях з робочим місцем, обладнаним ВДТ, має здійснюватись системою загального рівномірного освітлення. Як джерело штучного освітлення мають застосовуватись люмінесцентні лампи ЛБ.

Вимоги до освітлення приміщень та робочих місць під час роботи з ВДТ:

- освітленість на робочому місці повинна відповідати характеру зорової роботи, який визначається трьома параметрами: об'єктом розрізнення – найменшим розміром об'єкта, що розглядається на моніторі ПК; фоном, який характеризується коефіцієнтом відбиття; контрастом об'єкта і фону;
- необхідно забезпечити достатньо рівномірне розподілення яскравості на робочій поверхні монітора, а також в межах навколишнього простору;
- на робочій поверхні повинні бути відсутні різкі тіні;
- в полі зору не повинно бути відблисків (підвищеної яскравості поверхонь, які світяться та викликають осліплення);
- величина освітленості повинна бути постійною під час роботи;
- слід обирати оптимальну спрямованість світлового потоку і необхідний склад світла. Застосування світильників без розсіювачів та екрануючих ґратів заборонено.

Гігієнічні норми до організації і обладнання робочих місць з ВДТ. При розташуванні елементів робочого місця користувача ВДТ слід враховувати:

- робочу позу користувача;
- простір для розміщення користувача;
- можливість огляду елементів робочого місця;
- можливість ведення захистів;
- розміщення документації і матеріалів, які використовуються користувачем.

Конструкція робочого місця користувача ВДТ має забезпечити підтримання оптимальної робочої пози. Робочі місця з ВДТ слід так розташувати відносно вікон, щоб природне світло падало збоку, переважно зліва.

Робочі місця з ВДТ повинні бути розташовані від стіни з вікнами на відстані не менше 1,5 м, від інших стін — на відстані 1 м, відстань між собою – не менше ніж 1,5 м.

Принтер повинен бути розміщений у зручному для користувача положенні, так, що максимальна відстань від користувача до клавіш управління принтером не перевищувала довжину витягнутої руки користувача.

Конструкція робочого стола повинна забезпечувати можливість оптимального розміщення на робочій поверхні обладнання, що використовується, з врахуванням його кількості та конструктивних особливостей (розмір монітора, клавіатури, принтера, ПК та ін.) і документів, а також враховувати характер роботи, що виконується.

Вимоги до режимів праці і відпочинку при роботі з ВДТ. Під час роботи з ВДТ для збереження здоров'я працівників, запобігання профзахворюванням і підтримки працездатності встановлюються внутрішньо змінні регламентовані перерви для відпочинку.

Тривалість регламентованих перерв під час роботи з ЕОМ за 8-годинної денної робочої зміни залежно від характеру праці: 15 хвилин через кожен годину роботи – для розробників програм зі застосуванням ЕОМ; 15 хвилин через кожні

дві години – операторів із застосуванням ЕОМ; 10 хвилин після кожної години роботи за ВДТ для операторів комп'ютерного набору.

У випадках, коли виробничі обставини не дозволяють стосовувати регламентовані перерви, тривалість безперервної роботи з ВДТ не повинна перевищувати 4 годин.

Для зниження нервово-емоційного напруження, втомленості зорового аналізатора, для поліпшення мозкового кровообігу і запобігання втомі доцільно деякі перерви використовувати для виконання комплексу вправ, які передбачені ДСанПіН 3.3.2.007-98, в тому числі і для сеансів психологічного розвантаження у кімнаті з відповідним інтер'єром та кольоровим оформленням.

Виконання вимог ДСанПіН 3.3.2.007-98 повинне стати нормою всіх користувачів, які працюють над даним проектом.

#### 4.2 Забезпечення безпеки життєдіяльності при роботі з ПК

Робоче місце — це зона простору, що оснащена необхідним устаткуванням, де відбувається трудова діяльність одного працівника чи групи працівників.

Раціональне планування робочого місця має забезпечувати: найкраще розміщення знарядь і предметів праці, не допускати загального дискомфорту, зменшувати втомлюваність працівника, підвищувати його продуктивність праці. Площа робочого місця має бути такою, щоб працівник не робив зайвих рухів і не відчував незручності під час виконання роботи. Важливо мати також можливість змінити робочу позу, тобто положення корпусу, рук, ніг. Проте доцільно виключати або мінімізувати всі фізіологічно неприродні і незручні положення тіла.

Проведені дослідження показують, що при раціональній організації робочих місць продуктивність праці зростає знати на 15-25%.

Гігієнічні вимоги визначають умови життєдіяльності і працездатності людини у процесі взаємодії з технікою і середовищем; показниками є рівень освітлення, температура, вологість, шум, вібрація, токсичність, загазованість тощо.

Антропометричні вимоги визначають відповідність конструкцій техніки антропометричним характеристикам людини (зріст, розміри тіла та окремі рухові ланки). Показниками є раціональна робоча поза, оптимальні зони досягнення, раціональні трудові рухи.

Фізіологічні та психофізіологічні вимоги визначають відповідність техніки і середовища можливостям працівника щодо сприйняття, переробки інформації, прийняття і реалізації рішень.

Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- вибір ергономічно обґрунтованого робочого положення, виробничих меблів з урахуванням антропометричних характеристик людини;
- раціональне компонування обладнання на робочих місцях;
- урахування характеру та особливостей трудової діяльності.
- Загальні принципи організації робочого місця:
- на робочому місці не повинно бути нічого зайвого. Усі необхідні для роботи предмети мають бути поряд із працівником, але не заважати йому;
- ті предмети, якими користуються частіше, розташовуються ближче, ніж ті предмети, якими користуються рідше;
- предмети, які беруть лівою рукою, повинні бути зліва, а ті предмети, які беруть правою рукою — справа;
- якщо використовують обидві руки, то місце розташування пристосувань вибирається з урахуванням зручності захоплення його двома руками;
- робоче місце не повинно бути захарашене;



– організація робочого місця повинна забезпечувати необхідну оглядовість.

Статичні напруження працівника в процесі праці пов'язані з підтриманням у нерухомому стані предметів і знарядь праці, а також підтриманням робочої пози.

Робоча поза — це основне положення працівника у просторі: зручна робоча поза має забезпечувати стійкість положення корпусу, ніг, рук, голови працівника під час роботи, мінімальні затрати енергії та максимальну результативність праці.

Найпоширенішими у процесі праці є пози сидячи і стоячи. Проектуючи робоче місце, потрібно враховувати, що при виконанні роботи з фізичним навантаженням бажана поза стоячи, а при малих зусиллях — сидячи.

Робоча поза стоячи втомлює людину більше, ніж сидяча. Вона вимагає на 10 % більше енергії, спричиняє підвищення артеріального і венозного тиску крові, розширення вен на ногах, пошкодження ступень, викривлення хребта.

Під час роботи сидячи нижня частина корпусу розслаблена, а основне статичне навантаження припадає на м'язи шиї, спини, таза, стегон. Неправильна сидяча поза може викликати застій крові в ногах, а якщо виконується великий обсяг роботи для пальців рук — запалення суглобів.

#### 4.2.1 Параметри робочого місця

Загальні ергономічні вимоги для організації робочого місця користувача ПЕОМ (ГОСТ 12.2.049-80, ГОСТ 122032-78, ГОСТ 22269-76). Ці вимоги встановлюють основні параметри робочого місця, оснащеного дисплеєм, і враховують особливість виконуваних робіт.

Площа кабінету, в якому буде проходити робота повинна бути не менш 6 м<sup>2</sup>, а об'єм не менш 24 м<sup>3</sup>. Для внутрішньої обробки приміщення повинні

використовуватися дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі – 0,7-0,8; для стін – 0,5-0,6; для підлоги – 0,3-0,5.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання. Конструкція крісла повинна забезпечувати підтримку раціональної робочої пози під час роботи з відео-дисплейним терміналом (Далі ВДТ) і ПЕОМ, дозволяти змінювати позу з метою зниження статичного напруження м'язів шийно-плечової області і спини для попередження розвитку втоми працюючого (згідно з ГОСТ 12.2.032-78). Поверхня сидіння, спинки та інших елементів стільця (крісла) повинна бути напівм'якою, з покриттям, що не електризується, неслизьке та повітронепроникне, що забезпечує легке очищення від забруднення.

Висота робочої поверхні столу, за відсутності можливості її регулювання повинна складати 725 мм. Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною – не менше 500 мм, не менше 450 мм в глибину на рівні колін і на рівні простягнутої ноги – не менше 650 мм. Робоче місце має бути обладнане підставкою для ніг, має ширину не менше 300 мм, глибину не менше 400 мм, регулювання по висоті в межах 150 мм за кутом нахилу опорної поверхні підставки до 20 градусів.

Відстань від очей користувача до екрану дисплея має становити 500-700 мм. Кут зору 10-20°, але не більше 40°; кут між верхнім краєм дисплея і рівнем очей користувача має становити не менше 10°. Кращим є розташування екрану перпендикулярно до лінії зору користувача.

Робочі місця по відношенню до світлових прорізів повинні розташовуватися не ближче 3 м так, щоб природне світло падало збоку, переважно зліва. Освітленість також впливає на стан здоров'я і працездатність людини. У відповідності зі СНіП 11-4-79 встановлені наступні вимоги до освітленості:

Для штучного освітлення:

- Комбіноване освітлення – освітленість 1500 лк;

- Загальне освітлення – освітленість 400 лк.
- Для природного освітлення:
- Верхнє або комбіноване освітлення – коефіцієнт природної освітленості (далі КПО) 10%;
- Бічне освітлення – КПО 3.5%.
- Для суміщеного освітлення:
- Верхнє або комбіноване освітлення – КПО 3-6%;
- Бічне освітлення – КПО 1.1-2%.

До основних показників, що визначають умови здорової роботи, належать: фон, контраст об'єкта з фоном, видимість, показник осліпленості, коефіцієнт пульсації освітленості.

Фон характеризується коефіцієнтом відбиття. Контраст об'єкта з фоном (К) характеризується співвідношенням яскравості розглянутого об'єкта (точки, лінії, знаки) і фону. Оскільки роботи користувача ПЕОМ відносяться до категорії 1а – легкі фізичні роботи (роботи проводяться сидячи і супроводжуються незначним фізичним напруженням, з енерговитратами до 120 ккал / годину), необхідно дотримуватися наступних норм: коефіцієнт відображення більше 0,4, тобто світлий фон; контраст об'єкта з фоном великий і середній при К більше 0,2 (згідно СНіП 11-4-79).

У полі зору користувача ПЕОМ має бути забезпечений відповідний розподіл яскравості. Відношення яскравості екрана до яскравості оточуючих його поверхонь не повинно перевищувати у робочій зоні 3:1 (СНіП 11-4-79). У зв'язку з цим дисплей ПЕОМ повинен відповідати наступним вимогам:

- яскравість свічення екрану не менше 100 кд/м;
- мінімальний розмір світної точки для кольорового дисплея не більше 0,6 мм;
- контрастність зображення знаку – не менше 0,8;
- низькочастотне тремтіння зображення в діапазоні 0,05-1,0 Гц повинно знаходитися в межах 0,1 мм;

- екран повинен мати покриття антивідблиску;
- відеомонітор повинен бути обладнаний поворотним майданчиком, що дозволяє переміщати відеотермінал в горизонтальній і вертикальній площинах в межах 130-220 мм і змінювати кут нахилу на 10-15 мм.

Коефіцієнт відбиття світла матеріалами і обладнанням всередині приміщень має велике значення для освітлення: чим більше світла відбивається від поверхонь, тим вище освітленість. Коефіцієнт відображення відповідно повинен бути для: стелі 60-70%, стін 40-50%, підлоги 30%, для інших поверхонь 30-40%.

Результати досліджень показують, що найбільшою мірою негативний фізіологічний вплив на операторів ПК пов'язаний з дискомфорними зоровими умовами через неправильно спроектоване освітлення. Згідно СНіП II-4-79 освітленість на горизонтальній площині робочого місця оператора ЕОМ повинна складати 400 лк при висоті цієї площині 0,8 м над підлогою.

#### 4.2.2 Вимоги до освітленості і повітряного середовища в робочій зоні

Світловий клімат визначає зоровий дискомфорт. Запобігти шкідливому впливу освітлення можна шляхом правильного підбору системи освітлення, джерел світла (за їх спектрального складу випромінювання), світильників. Коли штучне світло змішується з природним, рекомендується використовувати лампи за спектральним складом найбільш близькі до сонячного світла. Світильники слід вибирати з розсіювачами, а блискучі деталі освітлювального обладнання, що можуть потрапити в поле зору оператора, повинні бути замінені на матові. Розташовувати робоче місце, обладнане дисплеєм, необхідно таким чином, щоб у полі зору оператора не потрапляли вікна або освітлювальні прилади; вони не повинні знаходитися і безпосередньо за спиною оператора. Вікна в приміщеннях з дисплеями обладнають шторами з коефіцієнтом відображення 0,5 ... 0,7, стіни

фарбують матовою фарбою з коефіцієнтом відображення 0,4 ... 0,6. Світловий клімат може бути поліпшений шляхом встановлення спеціальних антивідблискових контрастних фільтрів, однак при виборі типу фільтра необхідно враховувати умови роботи з комп'ютером, оскільки оптимальні значення коефіцієнтів пропускання і дзеркального відображення фільтрів залежать від освітленості робочого місця і типу джерела світла.

Враховуючи великий вплив освітлення на працездатність оператора при роботі з комп'ютером, проведемо розрахунок необхідної освітленості в приміщенні з дисплеями при наступних умовах: гігієнічна норма освітленості на горизонтальній поверхні на рівні робочого місця оператора – 400 лк; ширина приміщення – 7 м, довжина – 8 м, висота – 3 м. Коефіцієнт відбиття від стелі – 70, від стін – 50, від робочих поверхонь – 30. Повітряне середовище – нормальне (вміст пилу, диму й кіптяви не більше 5 мг/м<sup>3</sup>).

Повітряне середовище в робочій зоні визначається мікрокліматом виробничого приміщення. Величини температури, відносної вологості та швидкості руху повітря на робочих місцях з дисплеями повинні відповідати допустимим значенням, які встановлені ГОСТ 12.1.005-88 ССБТ для категорії робіт 1а (легкі фізичні роботи, вироблені сидячи і супроводжуються незначною фізичною напругою до 120 ккал/год.). Згідно з цим документом допустимі значення температури повітря в приміщенні становлять 19-25 °С, відносної вологості повітря – 55%, швидкості руху повітря на рівні особи – 0,1 м/с. При наявності досить комфортного робочого середовища атмосферний тиск по ГОСТ 21552-84 ССБТ може змінюватися від 84 до 107 кПа (630 ... 800 мм рт. ст.).

Шум несприятливий для людини, особливо при тривалому впливі. В оператора це виражається в зниженні працездатності (наприклад, швидкість обробки тексту зменшується на 10-15%), у прискоренні розвитку зорового стомлення, зміну відчуття кольору, підвищенні витрати енергії (на 17%). Тривалий та інтенсивний шум значно знижує продуктивність праці і призводить до зростання кількості помилок у роботі. У відділі головного економіста шум

може створюватися телефонними дзвінками та розмовами, системними блоками та клавіатурою ПЕОМ. Так само джерелами шуму можуть бути системи кондиціонування та вентиляування повітря, існують і зовнішні джерела шуму (наприклад, працюють агрегати на вулиці).

#### 4.2.3 Допустимі рівні звуку на робочих місцях

Допустимі рівні звуку та еквівалентні рівні звуку на робочих місцях повинні відповідати вимогам «Санітарних норм допустимих рівнів шуму на робочих місцях» № 3223-85. Згідно з цими нормами в приміщенні, де працює користувач ПЕОМ для забезпечення оптимальної робочої середовища рівень шуму не повинен перевищувати 60 дБ. Основними заходами боротьби з шумом згідно з ГОСТ 12.1.029-80 ССБТ є ліквідація або ослаблення джерела шуму шляхом застосування звукопоглинаючих матеріалів у конструкціях механізмів, використання коштів звукопоглинання і раціональна планування виробничого приміщення.

Випромінювання ПК можуть бути небезпечними для здоров'я. Низькочастотні поля при тривалому опроміненні сидять біля ПК людей можуть привести до порушень самих різних фізіологічних процесів. Відповідно до ГОСТ 27016-86 і ГОСТ 27954-88 потужність дози рентгенівського випромінювання в будь-якій точці простору на відстані 5 см від екрану відеомонітора при 41 годинному робочому тижні не повинна перевищувати 100 мкР/год (0,03 мкР/с), а інтенсивність ультрафіолетового випромінювання – 10 Вт/м<sup>2</sup>.

В даний час випускаються вибухобезпечні відеомонітори. За способом захисту людини від ураження електричним струмом дисплеї виготовляються відповідно з 1-м класом захисту за ГОСТ 25861-84, тому кабель живлення дисплея має вилку з трьома виводами, один з яких заземлюючий.

Для забезпечення ПДУ чинників робочого середовища на робочих місцях у необхідних випадках використовуються спеціальні засоби захисту працюючих. Способи захисту бувають активними і пасивними. Способи активного захисту засновані на виявленні джерел несприятливих факторів і вплив на них. У випадках неможливості здійснення активного захисту застосовується пасивна, за якої джерела несприятливих факторів залишаються, але здійснюються заходи, спрямовані на попередження вплив цих факторів на людину. Пасивна захист може бути колективного та індивідуального. Розглянемо колективні засоби захисту оператора ПК.

Висока температура повітря негативно позначається на функціональному стані людини. Всі основні електронні блоки ПК мають вбудовані вентилятори для забезпечення стабільних температурних режимів їх функціонування, тому при створенні комфортних умов роботи особливу увагу необхідно приділити шляхам відводу повітря (припливно-витяжної вентиляції). Для захисту від електростатичного потенціалу можуть бути використані згадані вище антиблисківі контрастуючі фільтри на екрани дисплеїв.

## ВИСНОВОК

Результатом магістерської роботи стало отримання повноцінної програмної системи у вигляді реалізації методів глибокого машинного навчання для вирішення задач навігації.

Перевагами даної системи є реалізована система для роботи з картографічними зображеннями, параметрами літальних об'єктів з врахуванням власних та зовнішніх параметрів. Система розроблена з використанням мови програмування та однойменного середовища MATLAB.

На етапі проектування було зроблено аналіз предметної області, обрано технологію розробки, визначено основні задачі та алгоритми для розробки методів машинного навчання.

На етапі розробки виконано програмний код як реалізацію методів глибокого машинного навчання для розв'язання задач навігації. Виконано підготовку тестових даних, тестування та навчання нейронної мережі. Під час процесу тестування не виявлено помилок та неточностей, відмічено повноту і якість вихідних даних, а також відповідність до вимог згідно ресурсів.

Даний програмний продукт виконує усі поставлені завдання, безпомилково обробляє вхідні дані та представляє результати роботи у зручному для користувача вигляді.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях», присвяченому основним аспектам безпеки праці, описуються правила безпеки під час написання програмного забезпечення. Оцінено основні ризики та наслідки порушень безпеки, описано основні правила для упередження надзвичайних ситуацій.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Christopher M. Bishop. Pattern Recognition and Machine Learning. - Springer, [Text] - 2006.
2. Le´on Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. - SIAM Review, [Text] 60:223--311, 2016.
3. Pablo Ramon Escobal. Methods of Orbit Determination. - Krieger Publishing Company, [Text] - 1965.
4. David Foster. Generative Deep Learning. - O’Reilly Media, Inc., [Text] - June 2019.
5. S. Haykin. Neural Networks. - Prentice-Hall, [Text] - 1999.
6. Stephanie Thomas and Michael Paluszek. MATLAB Machine Learning. - Apress, [Text] - 2017.
7. S. Russell and P. Norvig. Artificial Intelligence A Modern Approach Third Edition. Prentice-Hall, [Text] - 2010.
8. Жидецький В. Ц. Охорона праці користувачів комп’ютерів. – Львів: Афіша, 2000. - 176 с.
9. Правила безпечної експлуатації електроустановок споживачів [Текст] : ДНАОП 0.00-1.21-98. - Київ : Держнагляд охорони праці, 2003. - 383 с.
10. Наказ Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду «Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин» від 26.03.2010 № 65 – Режим доступу: URL: <http://zakon2.rada.gov.ua/laws/show/z0293-10>.
11. Марков В.В. Основа здорового способу життя профілактика хвороб: навч. посібник для студ. вищ. пед. навч. закладів. - М.: Академія, 2001. - 320 с.
12. М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – “Інженерія програмного забезпечення” для усіх форм навчання [Текст] –

Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя – 2020 – 27 с.

## **ДОДАТКИ**

## Додаток А

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
Факультет комп'ютерно-інформаційних систем і програмної інженерії  
Кафедра програмної інженерії

ЗАТВЕРДЖУЮ  
Завідувач кафедрою  
програмної інженерії

“ \_\_\_ “ \_\_\_\_\_ 2020 р.

### ТЕХНІЧНЕ ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра  
на тему: «Розробка методів глибокого машинного навчання для  
середовища MATLAB у розв'язанні задач навігації»

виконавець ст. гр. СПм-61  
Козачок Володимир Олексійович

\_\_\_\_\_  
(підпис)

керівник роботи:  
к. ф.-м. н., доц. Бойко Ігор Володимирович

\_\_\_\_\_  
(підпис)

Тернопіль 2020

## ЗМІСТ

### Вступ

1. Підстави до розробки
2. Призначення до розробки
3. Вимоги до програмного продукту
  - 3.1 Функціональні характеристики
  - 3.2 Склад та параметри технічних засобів
  - 3.3 Інформаційна та програмна сполучність
4. Стадії розробки
5. Програмна документація
6. Порядок контролю та приймання

## **1 ПІДСТАВИ ДО РОЗРОБКИ**

Розробка проводиться у відповідності до графіку навчального плану на 2020 рік, та згідно наказу на виконання кваліфікаційної роботи студента-магістра.

Тема проекту: «Розробка методів глибокого машинного навчання для середовища MATLAB у розв'язанні задач навігації».

## **2 ПРИЗНАЧЕННЯ РОЗРОБКИ**

Говорячи про новітні технології, можна з упевненістю сказати, що машинне навчання та штучний інтелект дедалі більше входять у всі сфери людського життя та спрощують багато складних процесів. Штучний інтелект та машинне навчання прискорюють роботу компаній, роблять їх ефективнішими, потужнішими, виводять на новий рівень.

Для реалізації ідеї, з врахуванням особливостей предметної області та коцепції машинного навчання в цілому, використано технологію та однойменну високорівневу мову програмування MATLAB. Важливим моментом є те, що тема є відносно новою і перспективною, що дозволяє застосувати максимальну кількість доступних інноваційних технологій та мати змогу впровадити систему у дію та доповнювати її відносно потреб у обраній сфері.

Об'єктом дослідження обрано актуальні питання обраної галузі, що дозволяє докладно вивчити проблематику, проаналізувати модливі способи вирішення з використанням передових методів програмування.

Як наукову новизну слід розглядати використання сучасних технологій машинного навчання для розробки унікального нового програмного забезпечення з урахуванням вимог та специфіки конкретної галузі.

Застосовувана сторона результатів наукових досліджень може бути конкретною діяльністю, і продукт повинен вдосконалюватись та адаптуватись до сучасних вимог.

### 3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

#### 3.1 Функціональні характеристики

Програмне забезпечення має виконувати наступні дії:

- надавати доступ до обробки вхідних даних та аналізу їх;
- працювати з тренувальними даними;
- спрощувати обробку вихідної інформації, передавати її в зрозумілому форматі для користувача;
- надавати графіки та числові дані передбаченої точності;

#### 3.2 Склад та параметри технічних засобів

- 1) ПК із 4096 Мб оперативної пам'яті, встановленою операційною системою Windows Seven, 8, 8.1, 10. Не менше 1024 Мб вільного місця на жорсткому диску. Двоядерний процесор з тактовою частотою від 1.2 GHz і більше.
- 2) Наявність встановленого MATLAB 9.5.

#### 3.3 Інформаційна та програмна сполучність

Програмний продукт повинен коректно функціонувати в операційних системах Windows Seven, 8, 8.1, 10, на яких доступний для встановлення MATLAB

3.4. Розроблювана система повинна бути пристосована для обробки тестових даних для тренування нейронної мережі, які повинні зберігатись у файловій системі ПК у вигляді сортованих файлів у каталогах.

#### **4. СТАДІЇ РОЗРОБКИ**

В ходів реалізації роботи проект повинен пройти крізь наступні стадії розробки:

- аналіз предметної області;
- аналіз доступних алгоритмів нейронних мереж;
- написання алгоритму нормалізації даних;
- проектування та розробка застосунку;
- проектування та розробка скриптів для роботи з зображеннями;
- оформлення супровідної документації;
- здача роботи.

#### **5. ПРОГРАМНА ДОКУМЕНТАЦІЯ**

Для програмного продукту повинні бути розроблені наступні документи:

- Пояснювальна записка;
- Технічне завдання;
- Презентаційний матеріал;
- Додатки.

#### **6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ**

Розроблений програмний продукт має виконувати всі вимоги, що складаються з перерахованих у п. 3.1 характеристик.

Приймання проводиться спеціально створеною екзаменаційною комісією в термін до: “\_\_” грудня 2020р.



**ДОДАТОК Б**

Публікація у науковому виданні

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ**

**МАТЕРІАЛИ**

**VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

**«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



**9–10 грудня 2020 року**

**ТЕРНОПЛЬ  
2020**

УДК 004.4

**В.О. Козачок**, ст. гр. СПМ-61

(Тернопільський національний технічний університет імені Івана Пулюя)

## **ВИКОРИСТАННЯ ГЛИБОКОГО МАШИННОГО НАВЧАННЯ У РОЗВ'ЯЗАННІ ЗАДАЧ ПРОГНОЗУВАННЯ**

UDC 004.4

**V.O. Kozachok**

## **THE USE OF DEEP MACHINE LEARNING IN SOLVING FORECASTING PROBLEMS**

Машинне навчання – великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися.

Штучний інтелект розвивається за допомогою алгоритмів прогресивного навчання і формує дані для подальшого використання. Він самостійно знаходить структуру та закономірності у даних та опрацьовує їх. Можливості такого навчання – безмежні з точки зору використання машин для вирішення широкого спектру задач. Моделі швидко адаптуються при отриманні нових даних, що поступово призводить до повного виключення помилок у реалізації певного автоматизованого процесу. Глибокий і ретельний аналіз дозволяє врахувати всі потенційні ризики, формує прогнози і попередження, виключає прийняття хибних рішень [1].

Для машинного навчання використовують різні технології та алгоритми. Все більше уваги почали приділяти штучним нейронним мережам, які, за визначенням, є системою з'єднаних і взаємодіючих між собою штучних нейронів, виконаних на основі порівняно простих процесорів. Кожен процесор штучних нейронних мереж періодично отримує сигнали від одних процесорів (або від сенсорів, або від інших джерел сигналів) і періодично посилає сигнали іншим процесорам. Всі разом ці прості процесори, з'єднані в мережу, здатні вирішувати досить складні завдання.

Глибоке навчання, як правило, чудово підходить для роботи з імовірнісними прогнозами за своєю природою. Причини цього, однак, не мають нічого спільного з проблемами ланцюга поставок. Алгоритми глибокого навчання добре підходять для імовірнісної / байєсівської оптимізації за допомогою таких метрик, як перехресна ентропія, оскільки такі метрики забезпечують великі значення градієнта, які добре поєднуються зі стохастичним градієнтним спуском, саме таким алгоритмом, що робить можливим глибоке навчання [2].

Підсумовуючи, глибоке машинне навчання може бути застосоване для багатьох задач з визначення закономірностей неструктурованих даних, таких як зображення, звук, відео чи текст. Методи глибокого машинного навчання можуть суттєво допомогти у створенні комп'ютерних систем, які виконують задачі, звично пов'язані з людським інтелектом.

### **Література.**

1. Штучний інтелект(ШІ): Що це таке і чому це так важливо? URL: <https://www.everest.ua/ai-platform/analytics/shtuchnij-intelekt-ai-shho-ce-take-i-chomu-ce-v/>.
2. Deep learning vs. machine learning in Azure Machine Learning. URL: <https://docs.microsoft.com/en-gb/azure/machine-learning/concept-deep-learning-vs-machine-learning> – Назва з екрану.

**ДОДАТОК В**