**ЗВІТ**

**Завдання**

Що таке нейромережа? Які типи мереж є, чим вони відрізняються, де використовуються. Основні фраймворки.

Нейромережа (Штучна нейронна мережа або Нейронка) — це математична модель, яка імітує структуру та функціонування біологічних нейронних мереж з метою вирішення різноманітних задач, таких як класифікація, регресія, прогнозування та генерація. Або ж нейромережа — це комп’ютерна система, яка намагається відтворити роботу людського мозку, а саме, як ми сприймаємо інформацію та вчимося. Нейронні мережі потрібні для вирішення складних задач, які важко розв’язати за допомогою класичних алгоритмів.

В основі нейромереж лежать штучні нейрони, які об’єднуються в графові структури і передають сигнали один одному через ваги зв’язків. Завдяки процесу навчання, під час якого ваги та зміщення між нейронами оптимізуються, нейромережі стають здатними до виявлення закономірностей та залежностей у вхідних даних.

Нейросери - це послідовність нейронів, з’єднаних між собою синапсами. Структура нейронної мережі прийшла у світ програмування прямо з біології. Завдяки такій структурі машина знаходить здатність аналізувати і навіть запам’ятовувати різну інформацію. Також нейронні мережі здатні як аналізувати вхідну інформацію, а й відтворювати її з власної пам’яті. Нейросери здатні самостійно навчатися і розвиватися, будуючи свій досвід на помилках.

Нейромережі активно використовуються в таких галузях:

* Класифікація: Визначення категорії, до якої належить певний об’єкт або подія, на основі його характеристик.
* Регресія: Прогнозування числового значення на основі вхідних даних.
* Генерація тексту: Створення тексту на основі навчальних даних, зазвичай використовується для автоматичного створення описів, статей або відповідей на запитання.
* Обробка зображень: Розпізнавання об’єктів, тексту або осіб на зображеннях.

Нейромережі знаходять широке застосування в різних галузях промисловості та сферах життя, таких як:

* Медицина: Діагностика захворювань, аналіз медичних зображень, передбачення ефективності лікування.
* Фінанси: Виявлення шахрайства, прогнозування цін на акції, оптимізація портфелів інвестицій.
* Маркетинг: Прогнозування вподобань клієнтів, автоматичне створення рекламних матеріалів, аналіз споживчої поведінки.
* Автоматичний переклад: Нейромережі можуть навчатися перекладати тексти між різними мовами, забезпечуючи швидкий та точний переклад.
* Розпізнавання мови: Розпізнавання та розуміння голосових команд для керування різними пристроями, такими як смартфони та домашня техніка.
* Автономні транспортні засоби: Навігація, ухилення від перешкод та безпечне керування автономними автомобілями або дронами.
* Безпека: Аналіз відео та зображень для виявлення підозрілої діяльності, захист від кібератак або виявлення шахрайства.
* Рекомендаційні системи: Аналізувати історію перегляду користувача та рекомендувати продукти, фільми, музику та інше на основі їх вподобань.

Типи нейронних мереж:

Нейронні мережі іноді описують з точки зору їхньої глибини. Саме тому термін “нейронна мережа” використовується майже як синонім терміну “**глибоке навчання**“. Їх також можна описати кількістю прихованих вузлів у моделі або кількістю входів і виходів, які має кожен вузол. Варіації класичного дизайну нейронної мережі дозволяють різні форми прямого і зворотного поширення інформації між рівнями.

**Конкретні типи штучних нейронних мереж включають**:

**Нейронні мережі прямого поширення** (Feed-forward neural networks): один з найпростіших варіантів нейронних мереж. Вони передають інформацію в одному напрямку, через різні вхідні вузли, поки вона не потрапить на вихідний вузол. Мережа може мати або не мати приховані шари вузлів, що робить її функціонування більш зрозумілим. Вона готова до обробки великої кількості шуму. Цей тип обчислювальної моделі ШНМ використовується в таких технологіях, як розпізнавання облич і комп’ютерний зір.

**Рекурентні нейронні мережі** (Recurrent neural networks): більш складні. Вони зберігають вихідні дані обчислювальних вузлів і подають результат назад у модель. Саме так модель навчається передбачати результат шару. Кожен вузол в моделі RNN діє як комірка пам’яті, продовжуючи обчислення і виконання операцій. Ця нейронна мережа починає з такого ж прямого поширення, як і мережа прямого поширення, але потім запам’ятовує всю оброблену інформацію, щоб повторно використовувати її в майбутньому. Якщо прогноз мережі невірний, то система самонавчається і продовжує працювати над правильним прогнозом під час зворотного поширення. Цей тип ШНМ часто використовується для перетворення тексту в мову.

**Згорткові нейронні мережі** (Convolutional neural networks – CNN): одна з найпопулярніших моделей, що використовуються сьогодні. Ця нейромережева обчислювальна модель використовує варіацію багатошарового сприйняття і містить один або кілька згорткових шарів, які можуть бути повністю з’єднані або об’єднані. Ці згорткові шари створюють карти ознак, які фіксують область зображення, яка в кінцевому підсумку розбивається на прямокутники і надсилається на нелінійний аналіз. Модель CNN особливо популярна у сфері розпізнавання зображень; вона використовується в багатьох найсучасніших додатках ШІ, включаючи розпізнавання облич, оцифрування тексту і обробку природної мови. Інші застосування включають виявлення перефраз, обробку сигналів і класифікацію зображень.

**Деконволюційні нейронні мережі** (Deconvolutional neural networks): використовують зворотний процес моделі CNN. Вони спрямовані на пошук втрачених ознак або сигналів, які спочатку вважалися неважливими для завдання системи CNN. Ця мережева модель може бути використана для синтезу та аналізу зображень.

**Модульні нейронні мережі** (Modular neural networks): містять кілька нейронних мереж, що працюють окремо одна від одної. Мережі не взаємодіють і не втручаються в роботу одна одної в процесі обчислень. Отже, складні або великі обчислювальні процеси можуть бути виконані більш ефективно.

До переваг штучних нейронних мереж належать:

* Здатність до паралельної обробки означає, що мережа може виконувати кілька завдань одночасно.
* Інформація зберігається у всій мережі, а не лише в базі даних.
* Здатність до навчання та моделювання нелінійних, складних взаємозв’язків допомагає моделювати реальні взаємозв’язки між входом і виходом.
* Відмовостійкість означає, що пошкодження однієї або декількох комірок ШНМ не зупинить генерацію вихідних даних.
* Поступове пошкодження означає, що мережа буде повільно деградувати з часом, замість того, щоб проблема знищила мережу миттєво.
* Здатність виробляти вихід з неповними знаннями, при цьому втрата продуктивності залежить від того, наскільки важливою є інформація, якої бракує.
* На вхідні змінні не накладається жодних обмежень, наприклад, як вони повинні бути розподілені.
* Машинне навчання означає, що ШНМ може вчитися на подіях і приймати рішення на основі спостережень.
* Здатність вивчати приховані взаємозв’язки в даних, не керуючись фіксованими зв’язками, означає, що ШНМ може краще моделювати дуже мінливі дані і непостійну дисперсію.
* Здатність узагальнювати і робити висновки про невидимі взаємозв’язки на основі невидимих даних означає, що ШНМ може передбачати вихід невидимих даних.

Недоліки нейронних мереж:

* Відсутність правил для визначення правильної структури мережі означає, що відповідна архітектура штучної нейронної мережі може бути знайдена лише шляхом проб і помилок та досвіду.
* Потреба в процесорах з можливістю паралельної обробки робить нейронні мережі апаратно-залежними.
* Мережа працює з числовою інформацією, тому всі проблеми повинні бути переведені в числові значення, перш ніж їх можна буде представити ШНМ.
* Відсутність пояснень, що стоять за пошуковими рішеннями, є одним з найбільших недоліків ШНМ. Неможливість пояснити, чому і як отримано рішення, породжує брак довіри до мережі.

Фреймворк у мовах програмування – це комплекс компонентів, бібліотек та інструментів, які пропонують структуру та готові рішення для роботи над певними завданнями. Фреймворки є готовими інструментами для розробки програмного забезпечення. Вони містять набір шаблонів, коду та інших елементів, що дозволяють розробникам скоротити час на написання коду та покращити його якість.

Базові переваги у використанні фреймворків :

* **Підвищення продуктивності та прискорення розробки**. Так як вони вже містять безліч готових рішень і шаблонів;
* **Поліпшення якості коду та супроводжуваності**. Фреймворки приділяють велику увагу цим аспектам;
* **Спрощення тестування та налагодження.** Готові рішення надають убудовані інструменти для цієї мети;
* **Поліпшення підтримки та можливість швидкої модифікації**. Фреймворки створені так, що їх зручно використовувати та модифікувати.

TensorFlow – це комплексна платформа для машинного навчання з відкритим вихідним кодом. Вона була розроблена командою Google Brain як продовження закритої системи машинного навчання DistBelief.

Як і більшість фреймворків глибокого навчання, TensorFlow має API на Python поверх механізму C та C++, що прискорює його роботу.

TensorFlow має гнучку екосистему інструментів, бібліотек та ресурсів спільноти. Це дозволяє дослідникам використовувати найсучасніші МО-технології, а розробникам — створювати та розгортати програми на базі машинного навчання.

Варто зазначити, що фреймворк постійно розвивається рахунок відкритого вихідного коду і великої спільноти ентузіастів. Також за рахунок його популярності є безліч вже вирішених завдань, що значно спрощує життя новоспеченим розробникам.

Однак фреймворк не позбавлений недоліків. Компанія Google відома своєю любов'ю до створення власних стандартів, що торкнулося і фреймворку. Наприклад, якщо під час роботи з TensorFlow у коді вилітає помилка, то фреймворк не покаже користувачеві конкретну сходинку, що її спровокувала.

PyTorch – це середовище машинного навчання мовою Python з відкритим вихідним кодом, що забезпечує тензорні обчислення з GPU-прискоренням. Фреймворк підходить для швидкого прототипування у дослідженнях, а також для аматорів та невеликих проектів.

Фреймворк пропонує динамічні графи обчислень, які дозволяють обробляти введення та виведення змінної довжини, що корисно, наприклад, при роботі з нейронними рекурентними мережами.

На відміну від TensorFlow, PyTorch менш гнучкий у підтримці різних платформ. Також у ньому немає рідних інструментів для візуалізації даних.

Однак, знову ж таки, на відміну від TensorFlow, якщо при роботі з PyTorch вилітає помилка, то це конкретна недоробка в коді і система виділить вам саме рядок, який її спровокував.

Keras – відкрите середовище глибокого навчання, написане на Python. Вона була розроблена інженером з Google.

Фреймворк націлений на оперативну роботу з нейромережами і є компактним, модульним і таким, що розширюється. Підходить для невеликих проектів, оскільки створити щось масштабне на ньому складно і він явно програватиме у продуктивності нейромереж того ж TensorFlow.

Фреймворк містить численні реалізації будівельних блоків нейронних мереж, що широко застосовуються, таких як шари, цільові та передавальні функції, оптимізатори, а також безліч інструментів для спрощення роботи із зображеннями і текстом.

Darknet - це фреймворк з відкритим вихідним кодом, написаний мовою C з використанням програмно-апаратної архітектури паралельних обчислень CUDA. Він швидкий, легкий та зручний у використанні. Також Darknet підтримує обчислення на базі CPU та GPU.

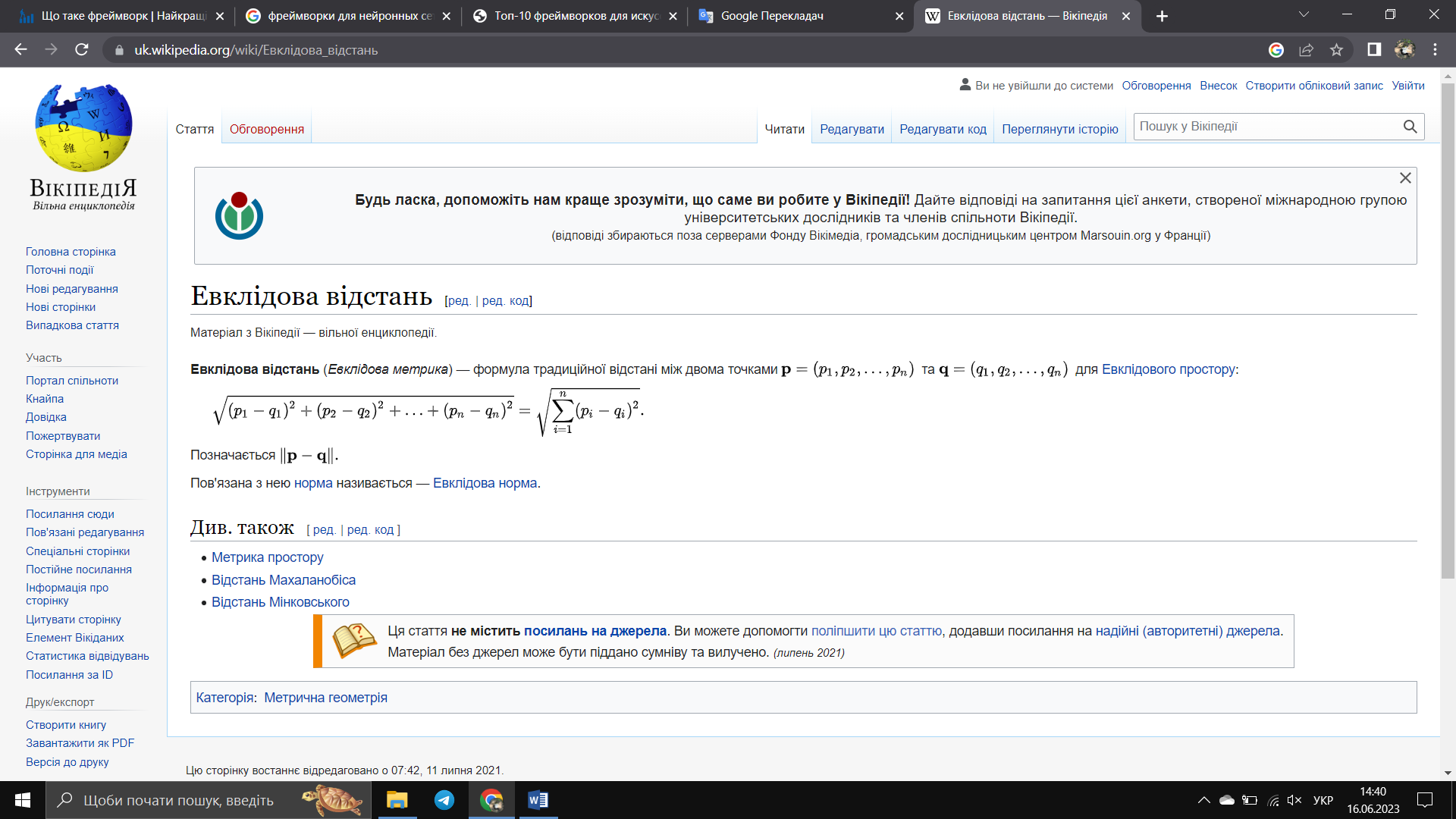
Darknet зберігає у форматі, який може бути розпізнаний за допомогою різних методів на різних платформах. Однак це може стати проблемою, якщо ви вирішите натренувати модель на одному надпотужному обладнанні, а потім використовувати її на іншому.

Так як фреймворк написаний на C і не має іншого API, то у випадку, коли вимоги платформи або власні переваги змусять звернутися до іншої мови програмування, вам доведеться заморочитися додатково над його інтеграцією. До того ж він поширюється лише у форматі вихідного коду, і процес компіляції на деяких платформах може бути проблематичним.

Фреймворк не рекомендується використовувати для складних проектів.

**Завдання**

Що таке евклідова відстань, вектор, вектор дескриптор?



Вектор — це стрiлка, яка однозначно визначається довжиною та напрямком.

Вектор має довжину (часто звану розмiром) i напрямок. Його можна iнтерпретувати як перемiщення в площинi. Вектор в програмуванні — це абстрактна модель, яка імітує динамічний масив.

Вектор являється контейнером, який містить послідовності елементів, і який може динамічно змінювати свій розмір. На подобі масивів, клас vector використовує послідовне скупчення елементів у пам'яті, що дозволяє виконувати арифметичні дії над вказівниками для доступу до його елементів.

Також, vector має власний вбудований клас ітераторів вільного доступу, над якими користувач може виконувати операції і які за функціоналом подібні до звичайних вказівників.

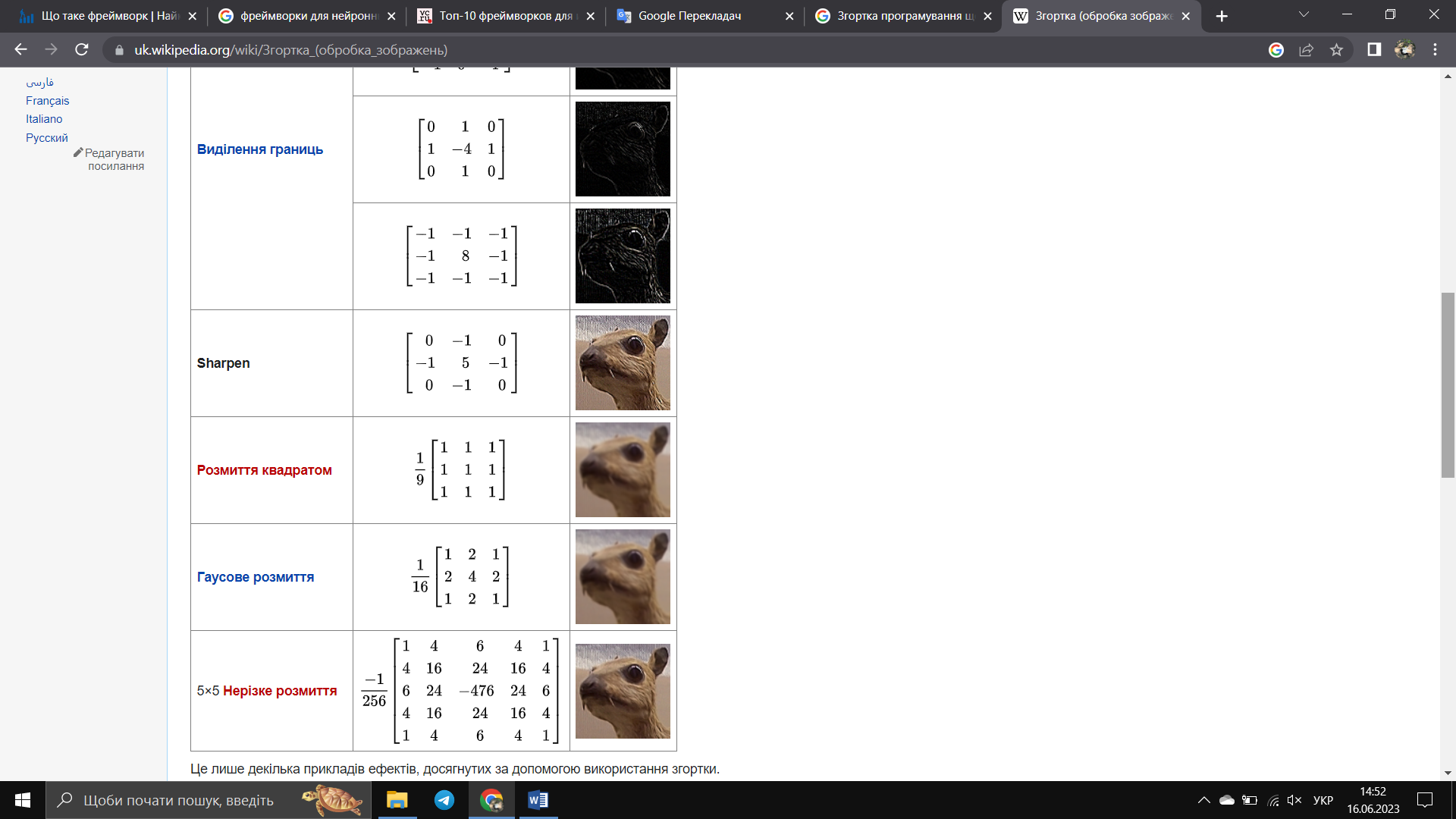
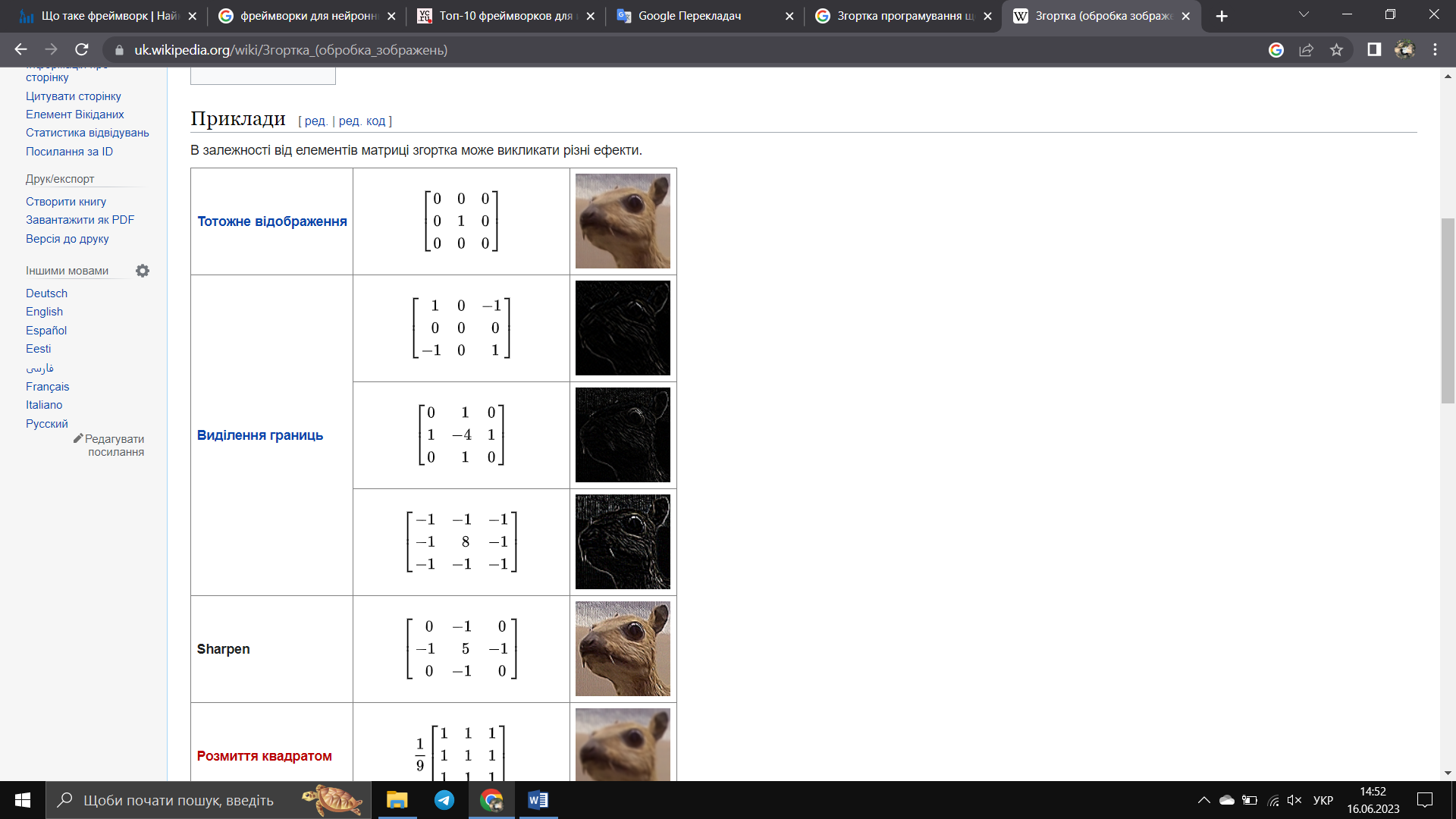
Це клас подібний на масив, який самостійно управляє пам’яттю, що звільняє програміста від рутинної роботи над виділенням пам’яті, її звільненням, переміщенням елементів і т.д. У порівнянні з іншими класами-контейнерами послідовностей (на подобі std:deque і std::list), std::vector має перевагу у швидкому доступі до його елементів.

[Дескриптори](https://docs.python.org/uk/3/glossary.html#term-descriptor) дозволяють об’єктам налаштовувати пошук атрибутів, зберігання та видалення. Векторний дескриптор— дескриптор, побудований на основі власних значень чи власних векторів.

**Завдання**

Згортка, що це таке, як працює, як використовується?

Згортка (ядро, матриця згортки) — матриця, зазвичай, малих розмірів, що використовується в оброблянні зображень як фільтр для розмиття, підвищення різкості, виділення границь тощо. Оброблення зображення полягає в обчисленні нового значення обраного пікселя з врахуванням значення навколишніх пікселів.



Згортка - це процес додавання кожного елемента зображення до його сусідів, зважених ядром. ([�������ℎ�]∗[123456789])[2,2]=(�⋅1)+(ℎ⋅2)+(�⋅3)+(�⋅4)+(�⋅5)+(�⋅6)+(�⋅7)+(�⋅8)+(�⋅9).

Значення кожного пікселя у вихідному зображенні рівне сумі добутків значень матриці згортки і відповідних пікселів вхідного зображення.

**Завдання**

Що таке згортки в нейронних мережах?

Згортка є одним з основних шарів у нейронних мережах для обробки даних, зокрема зображень. Вона використовується для локального виявлення функціональних особливостей у вхідних даних шляхом виконання оператора згортки між вхідними даними і набором фільтрів (які також називають ядром або фільтрами згортки).

Процес згортки полягає в послідовному переміщенні фільтра по вхідних даних та обчисленні зваженого сумарного значення пікселів у кожному регіоні, з яким фільтр перекривається. Отримані значення утворюють карту ознак, яка передається до наступного шару мережі. Фільтри в нейронній мережі навчаються під час процесу навчання з метою автоматичного виявлення певних шаблонів або ознак у вхідних даних.

Згортка має кілька важливих переваг:

* Локальне виявлення особливостей: завдяки переміщенню фільтра по вхідних даних, згортка може локально виявляти особливості, такі як ребра, кути, текстури тощо.
* Поділ параметрів: замість того, щоб мати окремий набір параметрів для кожного пікселя, згортка використовує спільний набір параметрів для фільтрів, що дозволяє ефективніше використовувати обмежені ресурси.
* Інваріантність до зсуву: згортка може виявляти особливості, незалежно від їх положення у вхідних даних.

**Завдання**

Що таке згорткові нейронні мережі?

Згорткові нейронні мережі - одні з найпопулярніших типів мережі, які часто використовуються для розпізнавання тієї чи іншої інформації у фотографіях та відео, обробці мови, системах для рекомендацій.

Згорткові нейронні мережі забезпечують часткову стійкість до змін масштабу, зсувів, поворотів, зміни ракурсу та інших спотворень. Ядро є фільтром, який ковзає по всьому зображенню і знаходить ознаки обличчя в будь-якому його місці (інваріантність до зсувів).

Ідея складної системи цього типу нейромережі виникла при ретельному вивченні зорової кори, яка у великих півкулях мозку відповідає за обробку візуальної складової. Основний критерій вибору користь згорткового типу – вона у складі технологій глибокого навчання.

Нам відомо, що нейронні мережі хороші у розпізнаванні зображень. Причому хороша точність досягається і звичайними мережами прямого поширення, проте, коли мова заходить для обробки зображень з великою кількістю пікселів, то кількість параметрів для нейронної мережі багаторазово збільшується. Причому настільки, що час, що витрачається на їхнє навчання, стає неймовірно більшим.

Головною особливістю згорткових мереж є те, що вони працюють саме із зображеннями, а тому можна виділити особливості, властиві саме їм. Багатошарові персептрони працюють з векторами, тому для них немає жодної різниці, чи знаходяться якісь точки поруч або на протилежних кінцях, так як всі точки рівнозначні і вважаються абсолютно однаковим чином. Зображення ж мають локальну зв’язність. Наприклад, якщо йдеться про зображення людських осіб, то цілком логічно очікувати, що точки основних частин тіла будуть поруч, а не розрізнено розташовуватися на зображенні. Тому потрібно знайти ефективніші алгоритми для роботи із зображеннями і ними виявилися згорткові мережі.

Основні характеристики :

* Відмінна масштабованість – проводять розпізнання образів будь-якого дозволу (яке б воно не було велике).
* Використання об’ємних тривимірних нейронів – усередині шару, нейрони пов’язані малим полем, іменовані рецептивним шаром.
* Механізм просторової локалізації – сусідні шари нейронів пов’язані таким механізмом, за рахунок чого забезпечується робота нелінійних фільтрів та охоплення дедалі більшої кількості пікселів графічного зображення.
* На відміну від мереж прямого поширення, що працюють з даними у вигляді векторів, згорткові мережі працюють із зображеннями у вигляді тензорів. Тензор — це 3D масиви чисел, або, простіше кажучи, масиви матриць чисел.

Зображення в комп’ютері відображаються у вигляді пікселів, а кожен піксель – це значення інтенсивності відповідних каналів. При цьому інтенсивність кожного каналу описується цілим числом від 0 до 255.

Найчастіше використовуються кольорові зображення, які складаються з RGB пікселів – пікселів, що містять яскравості по трьох каналах: червоному, зеленому та синьому. Різні комбінації цих кольорів дозволяють створити будь-який із кольорів всього спектра. Саме тому цілком логічно використовувати саме тензор для представлення зображень: кожна матриця тензора відповідає за інтенсивність свого каналу, а сукупність всіх матриць описує все зображення.