

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

МЕТОДИЧНІ ВКАЗІВКИ
до проведення практичних занять з дисципліни
**«МЕТОДИ ТА ЗАСОБИ ГЕНЕРАТИВНОГО
ІНТЕЛЕКТУ»**

для студентів
освітньо-професійної програми «Комп'ютерна інженерія»
спеціальності F7 «Комп'ютерна інженерія»
другого (магістерського) рівня вищої освіти

Тернопіль-2025

Березький О.М., Лящинський П.Б., Лящинський П.Б. Методичні вказівки до проведення практичних занять з дисципліни «Методи та засоби генеративного інтелекту» для студентів освітньо-професійної програми «Комп'ютерна інженерія» спеціальності F7 «Комп'ютерна інженерія» другого (магістерського) рівня вищої освіти. Тернопіль: ЗУНУ, 2025. 22 с.

Укладач: Олег БЕРЕЗЬКИЙ д.т.н., професор кафедри КІ
Петро ЛЯЩИНСЬКИЙ док. філософії, викладач кафедри КІ
Павло ЛЯЩИНСЬКИЙ аспірант кафедри КІ

Відповідальний за випуск: Дубчак Л.О., к.т.н., доцент, завідувач кафедри комп'ютерної інженерії

Рецензенти:

Микитишин А. Г. к.т.н., доц., завідувач кафедри комп'ютерно-інтегрованих технологій ТНТУ ім. Івана Пулюя
Манжула В.І. д.т.н., доцент кафедри комп'ютерних наук
Західноукраїнського національного університету

Методичні рекомендації рекомендовано до друку на засіданні кафедри
комп'ютерної інженерії
протокол № 7 від 28 лютого 2025 р.

ЗМІСТ

Практична робота №1. Створення зображень з використанням генеративно-змагальних нейронних мереж	4
Теоретичні відомості.....	4
Хід роботи.....	7
Зміст звіту	8
Контрольні запитання	9
Практична робота №2. Створення зображень з використанням Stable Diffusion (webui).....	10
Теоретичні відомості.....	10
Хід роботи.....	13
Зміст звіту	14
Контрольні запитання	15
Практична робота №3. Дослідження та аналіз ChatGPT	16
Теоретичні відомості.....	16
Хід роботи.....	19
Зміст звіту	19
Контрольні запитання	20
Список літератури.....	21

ПРАКТИЧНА РОБОТА №1

СТВОРЕННЯ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ ГЕНЕРАТИВНО-ЗМАГАЛЬНИХ НЕЙРОННИХ МЕРЕЖ

Мета роботи: дослідити процес генерування зображень за допомогою генеративно-змагальних нейронних мереж (GAN).

Теоретичні відомості

Генеративно-змагальні мережі (GAN) були запропоновані Яном Гудфеллоу у 2014 році [1]. Вони складаються із двох нейронних мереж, генератора G та дискримінатора D . Генератор намагається створити фейкові але водночас правдоподібні зображення, в той час як дискримінатор намагається відрізнити фейкові зображення (створені генератором) від реальних даних. Формально, генератор ставить у відповідність вектору шуму z , в певному латентному просторі, вхідне зображення: $G(z) \rightarrow x$, а дискримінатор визначається як $D(x) \rightarrow [0,1]$, що класифікує зображення як реальне (1) або фейкове (0).

Основна мета генеративно-змагальних мереж полягає у досягненні стану рівноваги, коли генератор настільки покращує якість своїх фейкових даних, що дискримінатор уже не здатен достовірно відрізнити їх від реальних даних. Це досягається шляхом оптимізації двох функцій втрат: генератор намагається мінімізувати помилку дискримінатора, у той час як дискримінатор прагне максимізувати свою точність. У математичному вираженні це виглядає як мінімакс-проблема, де обидві мережі навчаються одночасно.

Генератор приймає як вхід випадковий вектор шуму z , взятий із латентного простору, і перетворює його на зображення або інші типи даних, створюючи $G(z)$. Цей процес дозволяє досліджувати латентний простір, знаходячи різноманітні можливі варіанти даних, які відповідають заданому розподілу.

Дискримінатор, у свою чергу, приймає ці дані як вхід і видає значення в межах $[0, 1]$, де 1 означає, що дані реальні, а 0 — згенеровані.

GAN використовуються у багатьох сучасних застосуваннях, включаючи створення фотографій високої якості, генерування витворів мистецтва, перетворення стилів (наприклад, перехід між художнім і фотографічним стилями), створення реалістичних симуляцій та навіть у біомедицині для синтезу даних у випадках, коли реальних зразків недостатньо. Однак, попри їхню значну ефективність, GAN також мають певні виклики. Наприклад, процес навчання може бути нестабільним через складність досягнення рівноваги між генератором і дискримінатором. Крім того, генератор може схилитися до феномену «колапсу режиму», що означає створення лише обмеженого набору вихідних даних.

Таким чином, GAN є потужним інструментом, який продовжує розвиватися і знаходить дедалі більше застосувань у різних галузях, від розваг до науки, відкриваючи нові можливості у сфері штучного інтелекту.

Навчання GAN моделі передбачає пошук параметрів дискримінатора, що максимізує точність його класифікації, та пошук параметрів генератора, щоб максимально «обманути» дискримінатора. Навчання GAN моделі передбачає складний багатокроковий процес, у якому обидві мережі – генератор і дискримінатор – водночас навчаються та вдосконалюють свої функції. Генератор прагне створити зображення, які були б достатньо реалістичними, щоб дискримінатор помилково визначав їх як справжні. З іншого боку, дискримінатор постійно вдосконалює свою здатність розпізнавати згенеровані дані. Таке «змагання» між мережами дозволяє досягти високих результатів у створенні правдоподібних зображень. У процесі навчання використовується стохастичний градієнтний спуск, який дозволяє оптимізувати параметри кожної з мереж. Це вимагає ретельного налаштування гіперпараметрів, таких як кількість ітерацій, розмір міні-батчів і швидкість навчання, щоб забезпечити стабільність і ефективність процесу.

Алгоритм 1. Стохастичний градієнтний спуск для генеративно-змагальних мереж. Кількість кроків k , застосованих до дискримінатора, є гіперпараметром. $k = 1$.

for кількість ітерацій **do**

for k кроків **do**

- Вибрати міні-батч розміром m $\{z^{(1)}, \dots, z^{(m)}\}$ із розподілу $p_g(z)$;
- Вибрати міні-батч розміром m прикладів $\{x^{(1)}, \dots, x^{(m)}\}$ із початкового розподілу $p_{data}(x)$;
- Оновити параметри дискримінатора в напрямку збільшення градієнту:

$$\nabla \theta_d \frac{1}{m} \sum_{i=1}^m [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$$

end for

- Вибрати міні-батч розміром m $\{z^{(1)}, \dots, z^{(m)}\}$ із розподілу $p_g(z)$;
- Оновити параметри генератора в напрямку зменшення градієнту:

$$\nabla \theta_g \frac{1}{m} \sum_{i=1}^m [\log(1 - D(G(z^{(i)})))]$$

end for

Загалом принцип роботи будь-якої генеративно-змагальної мережі можна описати наступним чином. Дискримінатор виконує роль аналізатора, який має здатність розрізняти реальні дані від тих, які синтезовані генератором. Його основна мета — навчитися максимально точно класифікувати ці дані, використовуючи певну функцію втрат. У свою чергу, генератор прагне створити дані такої якості, щоб дискримінатор не зміг відрізнити їх від реальних, намагаючись заплутати дискримінатор.

Цей процес можна представити як гру із нульовою сумою між двома моделями, де кожна сторінка прагне покращити свої результати. Дискримінатор

максимізує імовірність класифікації реальних даних, мінімізуючи імовірність хибних класифікацій синтезованих даних. Генератор, навпаки, намагається мінімізувати функцію втрат, пов'язану із дискримінатором, щоб його синтетичні дані виглядали якнайбільш реалістично.

Цей підхід забезпечує двостороннє навчання: дискримінатор стає дедалі кращим у класифікації, а генератор — дедалі кращим у створенні синтетичних даних. У результаті, генеративно-змагальні мережі можуть бути використані для вирішення широкого спектра завдань, пов'язаних із синтезом даних — від створення реалістичних зображень до генерації складних структур даних, що можуть бути корисними у різних наукових і практичних контекстах.

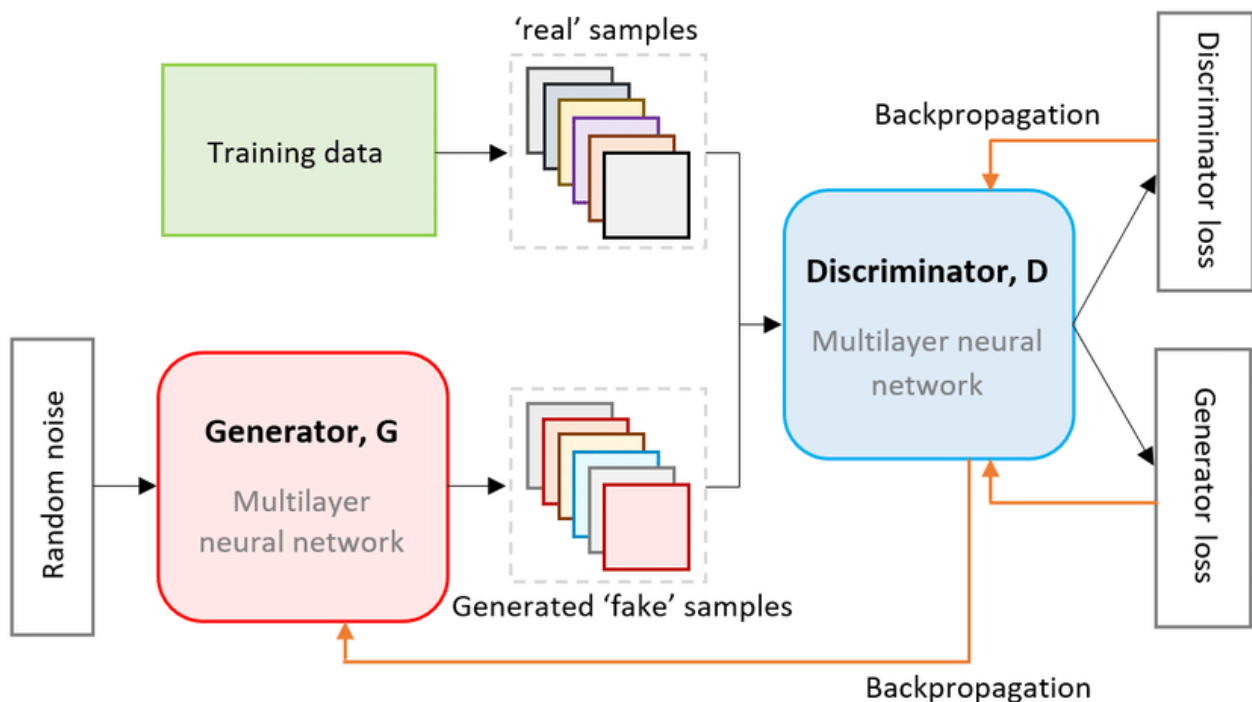


Рис.1. Архітектура GAN мережі

Хід роботи

Обрати варіант завдання відповідно до номера студента в журналі. Розробити генеративну модель GAN для створення зображень у наступних контекстах:

1. Створення простих геометричних фігур (датасет <https://www.kaggle.com/datasets/dineshpiyasamara/geometric-shapes-dataset>).
 2. Генерація зображень тварин (датасет <https://www.kaggle.com/datasets/andrewmvd/animal-faces>).
 3. Генерація зображень одягу (датасет <https://www.kaggle.com/datasets/zalando-research/fashionmnist>).
 4. Створення пейзажів (датасет <https://www.kaggle.com/datasets/arnaud58/landscape-pictures>).
 5. Генерація облич людей (датасет <https://www.kaggle.com/datasets/arnaud58/flickrfaceshq-dataset-ffhq>).
 6. Генерація рукописного тексту (датасет <https://www.kaggle.com/datasets/hojjatk/mnist-dataset>).
 7. Генерація автомобілів (датасет <https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset>).
 8. Створення фрагментів мистецьких полотен (датасет <https://www.kaggle.com/datasets/steubk/wikiart>).
 9. Генерація абстрактних зображень (датасет <https://www.kaggle.com/datasets/danielvalyano/abstract-paintings>).
 10. Створення зображень квітів (датасет <https://www.kaggle.com/datasets/imspars/flowers-dataset>).
- Приклад програмного коду наведено за посиланням:
<https://colab.research.google.com/drive/11B3cW8amCcSE5axX6rIXlXO6p710d45r?usp=sharing>

Зміст звіту

1. Титульна сторінка.
2. Тема та мета роботи.
3. Варіант завдання та код програми розв'язку.
4. Експериментальні результати (копії екранів роботи програми).

5. Висновки.

Контрольні запитання

1. Що таке GAN і з яких компонентів вона складається?
2. Як генератор перетворює шум на зображення?
3. Які гіперпараметри впливають на якість роботи GAN?
4. Які типи шарів використовуються в генераторі?
5. Що таке «режиму колапсу»?

ПРАКТИЧНА РОБОТА №2

СТВОРЕННЯ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ STABLE DIFFUSION (WEBUI)

Мета роботи: дослідити процес генерування зображень за допомогою інструменту Stable Diffusion та ознайомитися з його можливостями.

Теоретичні відомості

Stable Diffusion — це дифузійна модель, яка генерує зображення, трансформуючи шумовий сигнал у високоякісні графічні результати. Основу її роботи складає латентний дифузійний процес (Latent Diffusion Model, LDM), який суттєво оптимізує обчислювальні ресурси порівняно з класичними дифузійними моделями. Ця технологія дозволяє створювати реалістичні чи абстрактні зображення, використовуючи текстові описи або інші вхідні дані. Завдяки своїм можливостям Stable Diffusion знайшла застосування в різних сферах, таких як цифрове мистецтво, дизайн, маркетинг і навіть освіта. Однією з її ключових переваг є можливість адаптації під різні потреби користувачів — від простого створення ілюстрацій до складних творчих проєктів. Особливості роботи системи дозволяють генерувати зображення високої деталізації, а також легко налаштовувати параметри, такі як стиль, кольорова гама чи композиція результату.

AUTOMATIC1111 Stable Diffusion WebUI — це популярний інструмент із відкритим кодом, що надає веб-інтерфейс для роботи зі Stable Diffusion. Цей інструмент дозволяє користувачам максимально ефективно використовувати потужності дифузійної моделі, навіть якщо вони не мають глибоких технічних знань. Основними функціями є:

- Простий у використанні графічний інтерфейс: Завдяки інтуїтивно зрозумілому дизайну, інструмент підходить як для новачків, так і для

досвідчених користувачів. Він дозволяє легко запускати генерацію зображень на основі текстового опису або інших вхідних даних.

- Гнучке налаштування параметрів: Користувачі можуть змінювати такі параметри, як кількість ітерацій (Steps), шкала CFG (Classifier-Free Guidance Scale), розмір зображення, стиль і навіть рівень деталізації. Наприклад, значення CFG Scale можна встановлювати в діапазоні від 7 до 12 для оптимального балансу між точністю та художньою варіативністю.
- Можливість завантаження кастомних моделей: Інструмент підтримує різні версії моделей Stable Diffusion, включаючи модифіковані версії, оптимізовані для конкретних завдань, таких як створення фантастичних світів або реалістичних портретів.
- Розширений функціонал: WebUI включає такі інструменти, як:
- img2img: Дозволяє перетворювати існуючі зображення, додаючи до них нові деталі, стиль або навіть змінюючи їх формат.
- Inpainting: Ідеальний для виправлення або доповнення частин зображення. Наприклад, цей інструмент можна використовувати для заповнення пропущених областей або видалення небажаних елементів.
- LoRA (Low-Rank Adaptation): Ця функція дозволяє швидко адаптувати моделі для створення специфічних стилів або тем, без необхідності навчання з нуля.
- Вбудована інтеграція спільноти: Завдяки відкритому коду, користувачі мають можливість ділитися напрацюваннями, сценаріями генерації та модифікованими моделями, сприяючи розвитку екосистеми Stable Diffusion.

Цей інструмент знаходить застосування в багатьох сферах: від цифрового мистецтва, створення концептів для відеоігор і анімацій до маркетингових кампаній і освітніх проєктів. Для художників він відкриває нові горизонти творчості, а для компаній — нові способи візуальної комунікації. Завдяки своїм багатим можливостям, AUTOMATIC1111 Stable Diffusion WebUI стає

незамінним інструментом для всіх, хто прагне використовувати штучний інтелект у створенні графічного контенту.

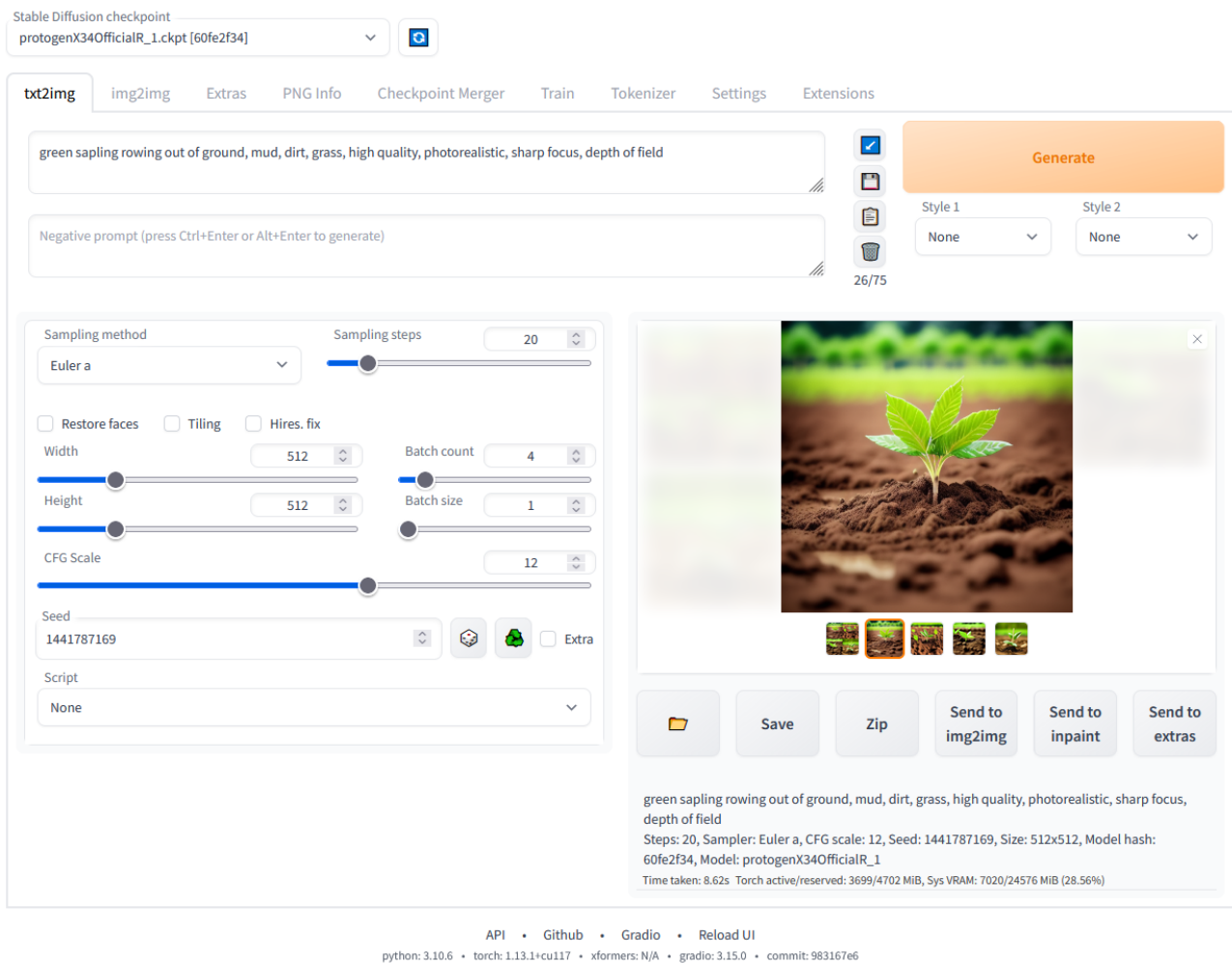


Рис. 1. Інтерфейс AUTOMATIC1111

Основні параметри генерації

1. Prompt (Підказка):

- Текстовий опис бажаного зображення. Якість підказки суттєво впливає на результат.

- Рекомендується використовувати детальні й контекстуальні описи.

2. CFG Scale (Classifier-Free Guidance Scale):

- Регулює баланс між точністю відповідності запиту й художньою варіативністю.

- Типові значення: від 7 до 12.

3. Steps (Кількість ітерацій):

• Визначає кількість етапів видалення шуму. Більше ітерацій зазвичай покращують деталізацію, але збільшують час генерації.

- Типовий діапазон: 20–50 кроків.

4. Розмір зображення:

• Генерація в стандартних розмірах (512x512 пікселів) оптимізована для моделі. Збільшення розміру може вимагати більше обчислювальних ресурсів.

Принцип роботи

1. Користувач вводить опис бажаного зображення (prompt).
2. Модель створює латентне представлення зображення на основі запиту.
3. Через визначену кількість кроків шум видаляється, і модель формує зображення.
4. Декодер перетворює латентне представлення на фінальне зображення.

Хід роботи

Обрати варіант завдання відповідно до номера студента в журналі. Ознайомитися з інструкціями по інсталяції програми локально за посиланням <https://github.com/AUTOMATIC111/stable-diffusion-webui/wiki> або використати хмарні сервіси <https://jarvislabs.ai/templates/automatic> (платно). Після встановлення програми згенерувати зображення відповідно до варіанту.

1. Створення простих геометричних фігур (датасет <https://www.kaggle.com/datasets/dineshpiyasamara/geometric-shapes-dataset>).
2. Генерація зображень тварин (датасет <https://www.kaggle.com/datasets/andrewmvd/animal-faces>).
3. Генерація зображень одягу (датасет <https://www.kaggle.com/datasets/zalando-research/fashionmnist>).

- | | | | |
|--|-----------|-------------|-----------------------------|
| 4. | Створення | пейзажів | (датасет |
| https://www.kaggle.com/datasets/arnaud58/landscape-pictures). | | | |
| 5. | Генерація | облич | людей (датасет |
| https://www.kaggle.com/datasets/arnaud58/flickrfaceshq-dataset-ffhq). | | | |
| 6. | Генерація | рукописного | тексту (датасет |
| https://www.kaggle.com/datasets/hojjatk/mnist-dataset). | | | |
| 7. | Генерація | автомобілів | (датасет |
| https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset). | | | |
| 8. | Створення | фрагментів | мистецьких полотен (датасет |
| https://www.kaggle.com/datasets/steubk/wikiart). | | | |
| 9. | Генерація | абстрактних | зображень (датасет |
| https://www.kaggle.com/datasets/danielvalyano/abstract-paintings). | | | |
| 10. | Створення | зображень | квітів (датасет |
| https://www.kaggle.com/datasets/imsparsh/flowers-dataset). | | | |

Експерименти:

- Вивчіть вплив зміни кількості ітерацій генерації (steps) на якість зображення.
- Експериментуйте зі зміною стилю текстових підказок (prompts).
- Налаштуйте параметри розміру зображення та оцініть, як це впливає на час генерації.

Зміст звіту

1. Титульна сторінка.
2. Тема та мета роботи.
3. Варіант завдання та код програми розв'язку.
4. Експериментальні результати (копії екранів роботи програми).
5. Висновки.

Контрольні запитання

1. Що таке дифузійні моделі, і які основні етапи їхньої роботи?
2. Які параметри впливають на якість результатів у Stable Diffusion?
3. Як Stable Diffusion оптимізує процес генерації зображень?
4. Чим дифузійні моделі відрізняються від GAN?
5. Які є переваги та недоліки використання Stable Diffusion?

ПРАКТИЧНА РОБОТА №3

ДОСЛІДЖЕННЯ ТА АНАЛІЗ CHATGPT

Мета роботи: Проаналізувати можливості великої мовної моделі (LLM) на прикладі ChatGPT, дослідити її обмеження та потенціал у задачах генерації тексту.

Теоретичні відомості

ChatGPT — це реалізація великої мовної моделі (LLM) GPT, побудованої на основі архітектури трансформерів, яка стала революційною технологією у сфері обробки природної мови (NLP). Ця модель використовує сучасні механізми генерації тексту, що забезпечують високий рівень розуміння контексту, дозволяючи створювати змістовні й точні відповіді на широкий спектр запитів.

GPT (Generative Pre-trained Transformer) складається з багатошарової архітектури трансформерів, що включає механізми самоуваги (self-attention) і лінійні шари. Навчання проходить у два етапи: етап переднавчання (pre-training), під час якого модель обробляє величезні обсяги текстових даних для формування загального розуміння мови, та етап донавчання з урахуванням специфічного контексту (fine-tuning), спрямованого на адаптацію моделі до конкретних завдань і запитів користувачів. Це дозволяє моделі бути універсальною і ефективною у вирішенні різних задач обробки природної мови, таких як текстова класифікація, машинний переклад, резюмування, генерація текстів тощо.

Однією з ключових особливостей GPT є механізм самоуваги, завдяки якому модель може оцінювати кожне слово тексту з точки зору його значущості у контексті інших слів. Це дозволяє аналізувати залежності між словами навіть у дуже довгих текстах і зберігати їхній контекст. У цьому процесі використовуються три основні компоненти: запити (Q), ключі (K) і значення (V). Вони застосовуються для обчислення вагів уваги (attention weights), які

визначають, наскільки одне слово впливає на інше в межах тексту. Завдяки цьому механізму, модель здатна розпізнавати навіть віддалені тематичні зв'язки між словами.

Механізм уваги є фундаментальною складовою архітектури трансформерів, який дозволяє моделі аналізувати текст і визначати залежності між словами на різних рівнях контексту. У процесі роботи цей механізм використовує три основні компоненти: запити (Q), ключі (K) та значення (V), що є векторами, які моделюють текстові дані.

Запити (Q) представляють поточне слово або токен, який аналізується, ключі (K) і значення (V) визначають взаємозв'язок між словами та їхньою значущістю в рамках контексту. Механізм уваги зводиться до обчислення вагів уваги, які показують, наскільки одне слово або токен впливає на інше. Ці ваги розраховуються через скалярний добуток запиту та ключа, нормалізуються з використанням softmax-функції, щоб отримати ймовірнісне розподілення. Потім отримані ваги множаться на значення, щоб визначити внесок кожного токена в остаточний представлення.

Окрім базового механізму, трансформери використовують багатоголову увагу (multi-head attention), що дозволяє моделі одночасно аналізувати текст із різних перспектив. Кожна "голова" відповідає за обробку окремого аспекту даних, наприклад, граматичних залежностей, семантичних зв'язків або контекстних нюансів. Це робить механізм уваги надзвичайно потужним в обробці складних текстових даних.

Увага також працює на різних рівнях: вона може бути спрямована як на окремі слова в межах речення (локальний контекст), так і на основні ідеї всього тексту (глобальний контекст). Завдяки цьому ChatGPT здатен точно розуміти зміст запиту, зв'язки між різними частинами тексту та формувати відповіді, які відповідають потребам користувача.

Крім того, GPT є універсальною завдяки її здатності обробляти багатомовні корпуси текстів, що дозволяє їй бути ефективною в різних мовних

середовищах. Це робить її корисною для широкого кола застосувань, таких як автоматизація клієнтської підтримки, створення контенту, рекомендаційні системи й навіть навчальні платформи. Наприклад, ChatGPT може використовуватися для перекладу текстів, написання художніх, технічних чи академічних матеріалів, створення сценаріїв та багато іншого.

Механізм уваги дозволяє ChatGPT не лише обробляти поточний запит, але й будувати складні, логічно пов'язані відповіді, враховуючи максимальну кількість деталей у запиті. Це дає можливість моделі відповідати на уточнюючі запитання, розширювати раніше надану інформацію чи пояснювати складні поняття простою мовою, роблячи технологію доступною навіть для не підготовлених користувачів.

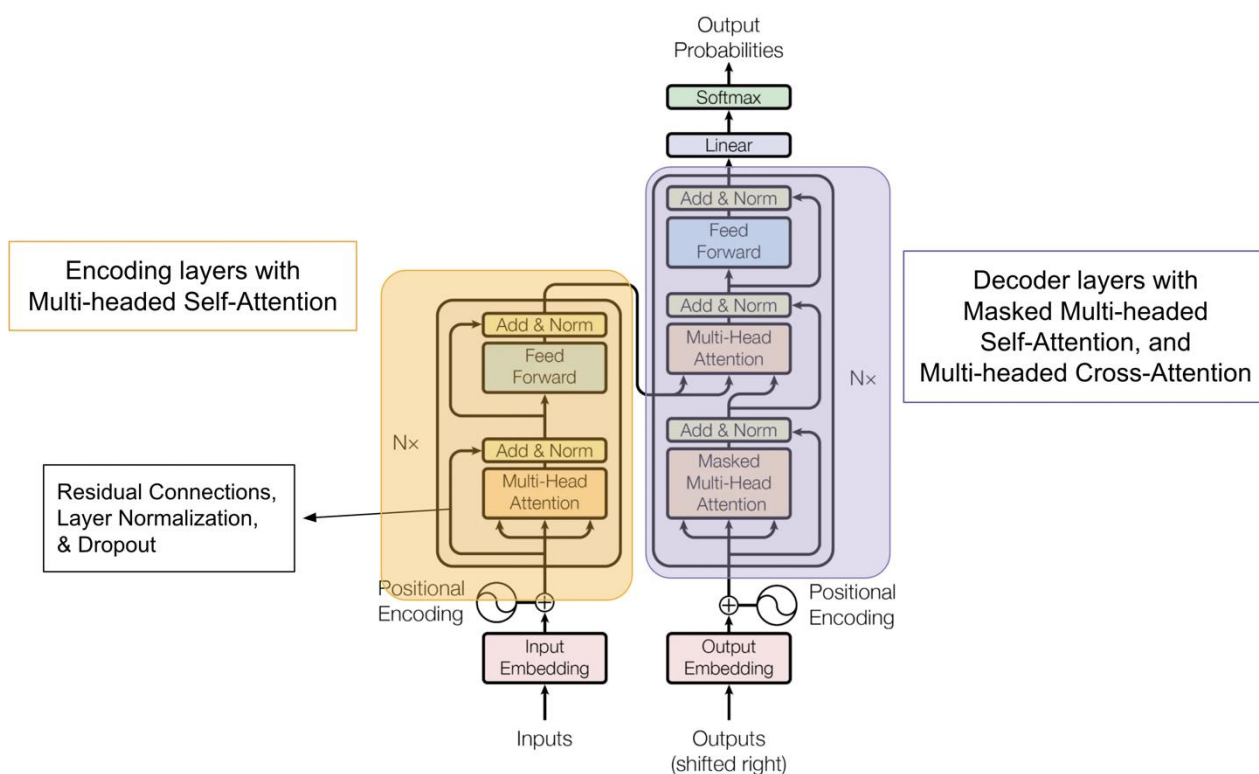


Рис. 2. Архітектура трансформера

Загалом, ChatGPT є прикладом технологічного прориву, що демонструє потенціал глибокого навчання у сфері обробки природної мови. Його успіх відкрив нові горизонти для реалізації штучного інтелекту в повсякденному

житті, науці та бізнесі, зробивши можливим виконання широкого спектра завдань із високим рівнем точності й адаптивності.

Хід роботи

Використовуючи ChatGPT, виконайте наступні завдання залежно від варіанту:

1. Написати коротку статтю на тему штучного інтелекту.
2. Перекласти текст із англійської на українську.
3. Створити резюме на задану тему.
4. Написати програму для розв'язання задачі обчислення Фібоначчі.
5. Скласти список рекомендацій для вивчення Python.
6. Згенерувати текст у стилі художнього роману.
7. Написати відповідь на запит користувача у форматі листа.
8. Пояснити складну тему простою мовою.
9. Створити список ключових слів для SEO.
10. Розробити короткий діалог між двома персонажами.

Експерименти:

- Проведіть тестування ChatGPT на різних рівнях складності запитів.
- Змініть стиль підказок і проаналізуйте, як це впливає на результати.
- Порівняйте відповіді ChatGPT на однакові запити різними мовами.

Зміст звіту

1. Титульна сторінка.
2. Тема та мета роботи.
3. Виконання індивідуального завдання (включаючи запит і результат).
4. Експериментальні результати (аналіз поведінки моделі).

5. Висновки.

Контрольні запитання

1. Які основні компоненти трансформера та яку роль вони виконують?
2. Як працює механізм уваги в GPT?
3. Чим відрізняється GPT від інших моделей, що базуються на трансформерах?
4. Як позиційне кодування впливає на обробку тексту?
5. У чому полягає унікальність багатоголової уваги?

СПИСОК ЛІТЕРАТУРИ

1. Generative AI Intro. URL: <https://roshni-mohandas.medium.com/gen-ai-introduction-5621b7e82134>
2. Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative ai. *Business & Information Systems Engineering*, 66(1), 111-126.
3. Generative Models Overview. URL: <https://paperswithcode.com/methods/category/generative-models>
4. Oussidi and A. Elhassouny, "Deep generative models: Survey," 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 2018, pp. 1-8.
5. Bank, D., Koenigstein, N., & Giryes, R. (2023). Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, 353-374.
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
7. From GAN to WGAN. URL: <https://lilianweng.github.io/posts/2017-08-20-gan/>
8. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29.
9. Iglesias, G., Talavera, E., & Díaz-Álvarez, A. (2023). A survey on GANs for computer vision: Recent research, analysis and taxonomy. *Computer Science Review*, 48, 100553.
10. Jabbar, A., Li, X., & Omar, B. (2021). A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys (CSUR)*, 54(8), 1-49.
11. Borji, A. (2019). Pros and cons of GAN evaluation measures. *Computer vision and image understanding*, 179, 41-65.
12. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33, 6840-6851.
13. Chan, S. H. (2024). Tutorial on Diffusion Models for Imaging and Vision. *arXiv preprint arXiv:2403.18103*.
14. How Diffusion models work. URL: <https://www.superannotate.com/blog/diffusion-models>
15. Stable diffusion. URL: <https://stable-diffusion-art.com/beginners-guide/>
16. Stable diffusion tutorials. URL: <https://www.stablediffusiontutorials.com/>
17. Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). Large language models: A survey. *arXiv preprint arXiv:2402.06196*.
18. Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
19. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
20. Introducing GPT. URL: <https://openai.com/index/chatgpt/>

21. GPT-3 architecture. URL: https://dugas.ch/artificial_curiosity/GPT_architecture.html
22. Fine-tuning. URL: <https://platform.openai.com/docs/guides/fine-tuning>
23. How GPT is developed. URL: <https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-language-models-are-developed>
24. Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., ... & Gadekallu, T. R. (2024). Gpt (generative pre-trained transformer)—a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*.
25. LLM fine-tuning. URL: <https://www.datacamp.com/tutorial/fine-tuning-large-language-models>
26. How to build a chat-bot. URL: <https://apiko.com/blog/how-to-build-an-ai-chatbot/>
27. Betzalel, E., Penso, C., Navon, A., & Fetaya, E. (2022). A study on the evaluation of generative models. *arXiv preprint arXiv:2206.10935*.
28. How to evaluate generative model. URL: <https://fastdatascience.com/generative-ai/how-can-we-evaluate-generative-language-models/>
29. Cloud AI. URL: <https://www.logicalcube.com/ai-cloud/overview/>
30. MLaaS. URL: <https://neptune.ai/blog/best-machine-learning-as-a-service-platforms-mlaas>
31. HuggingFace Learn. URL: <https://huggingface.co/learn>
32. OpenAI quickstart. URL: <https://platform.openai.com/docs/quickstart>
33. Learn PyTorch. URL: <https://www.learnpytorch.io/>
34. Berezsky, O., Liashchynskyi, P., Pitsun, O., Izonin, I. Synthesis of Convolutional Neural Network architectures for biomedical image classification *Biomedical Signal Processing and Control*, 2024, 95, 106325.
35. Berezsky O. M., Liashchynskyi P. B. Method of generative-adversarial networks searching architectures for biomedical images synthesis. *Radio Electronics, Computer Science, Control*. 2024. No 1, 104-117.
36. Лящинський, П. Б. (2024). Програмний засіб для класифікації та синтезу біомедичних зображень. *Scientific Bulletin of UNFU*, 34(4), 120-127. <https://doi.org/10.36930/40340415>.
37. Oleh Berezsky, Grygoriy Melnyk, Petro Liashchynskyi, and Oleh Pitsun Biomedical Image Datasets. *Lecture Notes on Data Engineering and Communications Technologies* this link is disabled, 2024, Springer, Cham.
38. Oleh Berezsky, Pavlo Liashchynskyi, Petro Liashchynskyi. Diagnostics methods and tools. Міжнародна науково-практична конференція «Обчислювальний інтелект і смарт-системи CISS-2024», 10 - 12 жовтня 2024 року, Львів, Україна.
39. Oleh Berezsky, Petro Liashchynskyi, Grygoriy Melnyk, Maksym Dombrovskyi and Mykola Berezkyi. Synthesis of biomedical images based on generative intelligence tools. *IDDM'24: 7th International Conference on Informatics & Data-Driven Medicine*, November 14 - 16, 2024, Birmingham, UK.