

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально–науковий інститут Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступень вищої освіти магістр

на тему «РОЗРОБКА МЕТОДУ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ
ТЕКСТОВОГО ОПИСУ З ВИКОРИСТАННЯМ ШТУЧНОГО
ІНТЕЛЕКТУ»

Виконав: студент 6 курсу, групи ПДМ - 62
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Сербулов Нікіта Євгенійович

(прізвище та ініціали)

Керівник

Трінтіна Н.А.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Магістр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ _____ ” _____ 2022 року

З А В Д А Н Н Я НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

СЕРБУЛОВА НІКІТИ ЄВГЕНІЙОВИЧА

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка методу генерації зображень на основі текстового опису з використанням штучного інтелекту»

Керівник роботи: Трінтіна Н.А. к.т.н., доц., доцент кафедри ІІЗ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» жовтня 2022 року №122.

2. Строк подання студентом роботи «30» грудня 2022 року

3. Вхідні дані до роботи

Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо розробки та проведення математичних досліджень

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Аналіз наявних математичних методів та технологій для розробки систем аналізу тональності тексту.

4.2 Дослідження інформаційних технологій за темою.

4.3 Реалізація інформаційної системи для аналізу тональності тексту.

4.4 Опис проектування системи.

4.5 Опис використаних технологій.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Титульний слайд
 2. Мета, об'єкт та предмет дослідження
 3. Актуальність роботи
 4. Аналіз існуючих рішень
 5. Результати дослідження
 6. Апробація результатів дослідження
 7. Висновки
6. Дата видачі завдання «14» жовтня 2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Вивчення теми магістерської роботи	14.10.2022	Виконано
2	Вивчення літературних джерел	17.10.2022	Виконано
3	Складання плану роботи	22.10.2022	Виконано
4	Узгодження плану роботи та списку використаних джерел з науковим керівником	01.11.2022	Виконано
5	Аналіз існуючих математичних рішень	03.11.2022	Виконано
6	Дослідження інформаційних технологій	07.11.2022	Виконано
7	Розробка модифікованої математичної моделі та проведення дослідження	13.11.2022	Виконано
8	Вступ, висновки, реферат	03.12.2022	Виконано
9	Розробка обов'язкових демонстраційних матеріалів	12.12.2022	Виконано
10	Попередній захист роботи	22.12.2022	Виконано
11	Здача роботи	18.01.2022	

Студент _____
(підпис)

Сербулов Н. Е.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Трінтіна Н. А.
(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 66 с., 16 рис., 12 фор., 1 табл. , 20 джерел.

Об'єкт дослідження – моделювання поведінки штучного інтелекту.

Предмет дослідження – дослідження моделі штучного інтелекту для пришвидшення роботи його алгоритмів

Мета роботи – пришвидшення процесу отримання та аналізу даних штучного інтелекту для безперервного навчання з метою підвищення ефективності побудованих моделей.

Результати проведеного наукового дослідження дають зрозуміти, що використані математичні методи, алгоритмічні моделі, та програмні засоби надають можливість вдосконалити процес отримання та аналізу даних штучного інтелекту.

Практичні значення отриманих результатів полягає в тому, що на основі проведених теоретичних досліджень було розроблено нову метод обробки тексту на основі автоенкодингу у поєднанні з методами машинного навчання та обробки зображень. Результати цього дослідження надалі дають змогу пришвидшити роботу алгоритмів штучного інтелекту

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. МАШИННЕ НАВЧАННЯ В АНАЛІТИЦІ	12
1.1 Загальні поняття.....	12
1.1.1 Поняття машинного навчання у штучному інтелекті.....	12
1.1.2 Модель алгоритму та її вибір.....	13
1.2 Робота з даними.....	13
1.2.1 Види даних для роботи з машинним навчанням.....	13
1.3 Основні завдання машинного навчання в аналітиці.....	16
1.3.1 Класифікація, регресія, прогнозування.....	16
1.3.2 Кластеризація.....	18
РОЗДІЛ 2. ОЦІНКА ЯКОСТІ ПОБУДУВАННЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ	19
2.1 Критерії оцінювання моделі.....	19
2.2 Основні причини помилок у прогнозах.....	20
2.3 Методи валідації моделей.....	25
2.4 Порівняльний аналіз моделей.....	29
РОЗДІЛ 3. РОЗРОБКА КОРИСТУВАЛЬНОГО ІНТЕРФЕЙСУ ДЛЯ АВТОМАТИЗАЦІЇ ПОБУДУВАННЯ ТА ВИКОРИСТАННЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ	31
3.1. Розробка механізму конфігурації моделей машинного навчання.....	31
3.2. Розробка автоматичного порівняння запропонованих конфігурацій моделей.....	32
3.3. Безперервне перенавчання побудованої моделі.....	33
3.4. Актуальність запропонованого рішення для роботи аналітика та бізнесу.....	34
3.5. Ефективність запропонованого рішення для роботи аналітика та бізнесу.....	35
3.6 Генеративний ШІ.....	49
3.7 Автоенкодера.....	56
3.8 Векторні квантовані генеративні змагальні мережі (VQ-GAN).....	58
3.9 Проведення моделювання за допомогою коду.....	60

ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

III	–	Штучний інтелект
ISO	–	International Organization for Standardization
OpenAI	–	Лабораторія досліджень штучного інтелекту
DALLE	–	DALL-E and DALL-E 2 are deep learning models developed by OpenAI to generate digital images from natural language descriptions, called "prompts".
GPT-3	–	Generative Pre-trained Transformer 3

ВСТУП

З розвитком комп'ютерних технологій все більш актуальним стає питання вдосконалення та спрощення роботи з комп'ютерним кодом. В даний час комп'ютери можна не просто програмувати, а й налаштовувати їх так, щоб вони навчалися самі. Ця технологія полягає в оцінці раніше зроблених процесів і припущенні, які б дії зробив людина у цій ситуації. У цьому існують критерії, якими можна зрозуміти, наскільки правильно прийняв рішення алгоритм. Так як алгоритмів машинного навчання безліч, саме ці критерії і є основним показником, за яким алгоритми та моделі порівнюють між собою.

Чим більше алгоритм прикладів для машинного навчання, тим більше точно він зможе робити передбачення. Це створює першу проблему - яким чином з багатьох видів даних, зібрати і призвести до потрібного формату необхідну інформацію для подальшого навчання? Для цього необхідні фахівці, які мають кваліфікацію та здатні використовувати навички програмування на практиці. Це відразу створює другу проблему - яким чином аналітикам, зокрема економістам, використовувати алгоритми машинного навчання для своїх цілей, якщо вони не володіють навичками програмування і не зможуть виконувати технічно складні дії для аналізу електронних даних. Було проаналізовано результати запропонованого рішення. У результаті було виявлено, що немає єдиного рішення для побудови моделі. Аналітики постійно спостерігають за різними сплесками в економічному середовищі, внаслідок чого побудовані раніше моделі стають неефективними. Крім того, великі компанії постійно працюють над створенням більш досконаліх алгоритмів, використовуючи та впроваджуючи нові інструменти машинного навчання. Основною проблемою для бізнесу є вибір відповідної моделі, а також її валідацією та безперервним оновленням на свіжих даних.

У цій роботі будуть розглянуті основні положення з теорії машинного навчання, будуть порушені основні проблеми збору та аналізу даних, а також буде запропоновано вирішення всіх раніше перерахованих проблем.

Мета роботи: пришвидшення процесу отримання та аналізу даних штучного інтелекту для безперервного навчання з метою підвищення ефективності побудованих моделей.

Об'єкт дослідження: моделювання поведінки штучного інтелекту.

Предмет дослідження: дослідження моделі штучного інтелекту для пришвидшення роботи його алгоритмів

Використовувані методи та інструментарій. Для побудови моделей використовуються декілька найпопулярніших алгоритмів градієнтного бустингу: Xgboost, Catboost, LightGBM. Для обробки даних використовується мова програмування python, JavaScript. Інтерфейс користувача представляє web додаток на основі Java, JavaScript, MySQL, ExistDB.

РОЗДІЛ 1. МАШИННЕ НАВЧАННЯ В АНАЛІТИЦІ

1.1 Загальні поняття

1.1.1 Поняття машинного навчання у штучному інтелекті

Машинне навчання - це область штучного інтелекту, що розглядає спосіб розробки алгоритмів, з можливістю навчання. Цей термін було запропоновано А. Л. Самуєєм наприкінці 1950-х років, винахідником першої комп'ютерної програми для гри в шашки, яка могла навчатися. Так звучить перше формулювання:

Машинне навчання (Machine learning) — процес, який дає можливість комп'ютерам навчатися виконувати щось без явного написання коду.

Машинне навчання лежить на стику традиційної математики та математичної статистики. Крім того вирішує проблеми, пов'язані з перенавчанням і точності та обчислень. Основний перелік методів машинного навчання проектувався заміни традиційних статичних підходів. Майже всі способи стосуються отримання даних з подальшим аналізом. Деякі сегменти машинного навчання віднесені до спеціального напрямку і називають доктриною обчислювального навчання.

Машинне навчання має широке застосування на практиці, а не лише аналітичний та теоретичний зміст. Перевірка різних методів та алгоритмом на реальних даних досягається за рахунок емпіричного досвіду. Для досягнення ефективної роботи запропонованого рішення застосовуються додаткові методи, які відображають помилки прогнозних значень від реальних даних.

Для оцінки практичної користі алгоритму використовують різні експерименти на прогнозних значеннях і реальних даних

1.1.2 Модель алгоритму та її вибір

Вирішальне правило - така функція для фіксованого значення ваг, яка містить у собі:

- Навчальну вибірку, що містить безліч ознак
- Безліч додаткових ваг функції

Модель алгоритму - загальний обсяг вирішальних правил, що надають різні ваги всім можливим значенням.

Завдання пошуку найкращого алгоритму ділиться на дві частини:

- Налаштування гіперпараметрів (параметри, які задаються заздалегідь і не змінюються по ходу навчання)
- Завдання вибору найкращого алгоритму з відомих

Вибір моделі може відбуватися з множини за допомогою ручного тестування. Головна проблема цього методу – обмежена кількість можливих моделей, які можна перевірити, тому зазвичай використовують автоматичні алгоритми вибору моделей.

1.2 Робота з даними

1.2.1 Види даних для роботи з машинним навчанням

Більшість алгоритмів машинного навчання потрібен великий обсяг даних навчання. Крім цього, необхідно порівнювати також ефективність збудованих моделей. Так як пошук і розмітка даних складне завдання, що вимагає великих витрат, для тестування моделей використовують вже зібрані набори даних,

приведені в зручний формат. Для більшості цих наборів вже опубліковані результати роботи різних відомих алгоритмів, тому можна зробити оцінку, наскільки ефективно працює модель.

Розглянемо найпопулярніші набори даних:

- Iris

Це зовсім невеликий набір даних, який використовується для вирішення задач класифікації. Набір був опублікований у 30-ті роки ХХ століття, в основі лежать дослідження американського ботаніка Едгара Андерсона. Набір містить опис трьох видів квіток. У кожній квітці було виміряно основні числові показники (такі як довжина, ширина тощо), а потім вони були позначені своїм класом.

Длина чашелистника	Ширина чашелистника	Длина лепестка	Ширина лепестка	Клас
5.1	3.5	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.3	3.3	6.0	2.5	virginica

Рисунок 1.1 – Приклад набору даних “IRIS”

Такі дані дозволяють швидко перевірити алгоритми класифікації.

- MNIST

Досить великий відносно попереднього набору. Дозволяє перевірити об'єкт на приналежність одного з 10 класів: цифри із десяткової системи числення. Містить близько шістдесяти тисяч тренувальних елементів, а також десять тисяч

перевірочних, які позначені певним класом (числом). Є набір картинок з рукописними цифрами. Також використовується для перевірки алгоритмів, які вирішують завдання класифікації.

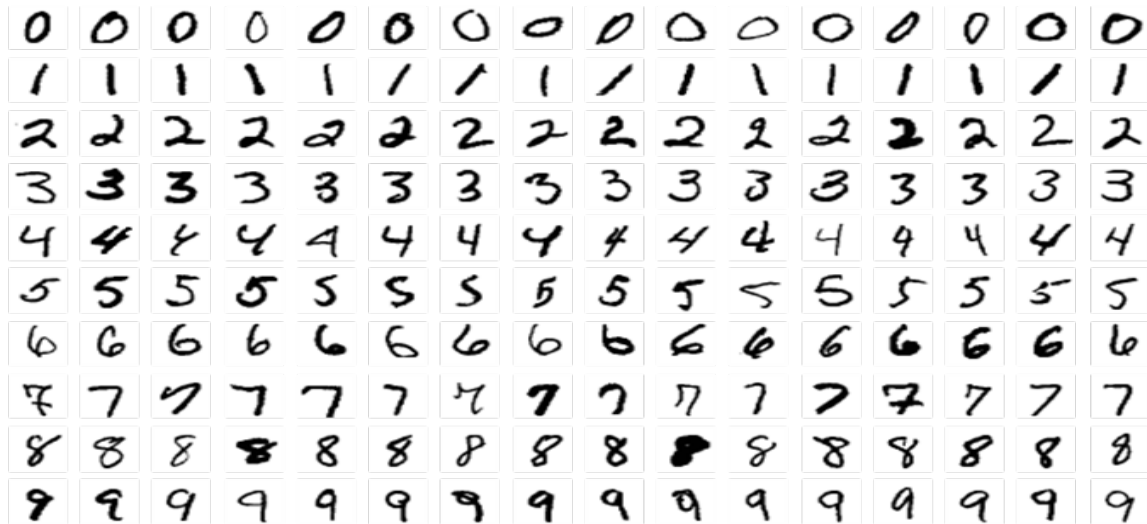


Рисунок 1.2 – Приклад набору даних “MNIST”

- CIFAR-10

Дані для тестування алгоритмів машинного навчання, що являють собою набір зображень. Усі зображення розбиті на класи за вмістом: різні тварини, техніка тощо. Кожен клас містить близько шести тисяч кольорових зображень. На даний момент алгоритми можуть класифікувати зображення з точністю 148%.



Рисунок 1.3 – Приклад набору даних “CIFAR-10”

- ImageNet

База зображень, у якій до кожного об'єкта прописані інструкції. Відмінною рисою з інших баз і те, що у одному зображенні може бути кілька розпізнаних об'єктів. Анотації додаються звичайними користувачами, що дозволяє збільшити кількість даних.

1.3 Основні завдання машинного навчання в аналітиці

1.3.1 Класифікація, регресія, прогнозування

Навчання з учителем - основний сегмент у машинному навчанні, який вирішує проблему в умовах великої кількості об'єктів та різних варіантів відповіді. Кінцевою метою є завдання пошуку зв'язку між відповідями та об'єктами. Навчальна вибірка – сукупність прецедентів (об'єкт – відповідь). Зазвичай під учителем мають на увазі саму навчальну вибірку, або того, хто вказав відповіді заданим об'єктам. Для побудови алгоритму потрібно знайти залежність між безліччю об'єктів та безліччю відповідей, такий алгоритм прийматиме об'єкт на

вході та видаватиме на виході відповідь. Кожен прецедент є парою "об'єкт, відповідь". Функціонал якості визначаємо як середня помилка відповідей, які видає алгоритмом по всіх об'єктах вибірки. На основі навчання з учителем вирішуються багато завдань, розглянемо основні з них:

- Завдання класифікації

Щоб створити навчальну вибірку, візьмемо кілька об'єктів і розділимо їх на класи. При цьому відомо, яким конкретно класам відносяться об'єкти. Класова приналежність інших об'єктів невідома. З усього цього безлічі з допомогою побудови алгоритму можна класифікувати (тобто вказати номер чи найменування класу, якого даний об'єкт належить) будь-який довільний об'єкт. Таким чином, ми отримуємо номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування до конкретного об'єкта.

- Завдання регресії

У задачах регресії, можливим результатом може бути числовий вектор, що відрізняє проблему регресії від класифікаційних завдань. Датасет має структуру ключа - значення. Де ключем є залежна змінна, а значення використовується незалежна змінна. Як регресійну модель використовують функцію, яка відображає набір вхідних даних та прогнозне значення. Щоб модель показувала якісний результат, потрібно грамотно підібрати гіперпараметри. Середньоквадратична помилка часто використовується для визначення точності прогнозів моделі та розраховується як сума квадратів різниці прогнозного та реального значень. Завдання регресії застосовується для аналізу часових рядів та виділення зв'язків між змінними.

- Завдання прогнозування

Відрізки часових рядів вважаються об'єктами. Дані відрізки перериваються, коли потрібно дати відповідь значення у майбутньому. Методи класифікації та регресії також можна використовувати у задачах прогнозування. Завдання прогнозування також можна подати у вигляді прогнозу значень тимчасового ряду,

грунтуючись на попередніх результатах та необхідною метою інформації. Наприклад, під такою інформацією можна розглянути зовнішні умови. Так, політична ситуація може позначитися на курсі акцій, мита можуть вплинути на експорт у країні.

1.3.2 Кластеризація

У навчанні без вчителя виявляються внутрішні взаємозв'язки та закономірності об'єктів, оскільки з його вирішуються завдання обробки даних, у яких відома лише навчальна вибірка (опис безлічі об'єктів). Тобто. Тренувальні дані доступні все відразу, але відповіді для поставленого завдання невідомі.

Завдання, які можуть вирішуватись цим способом: кластеризація, знаходження асоціативних правил, видача рекомендацій (наприклад, реклама), зменшення розмірності датасета. Розглянемо найпопулярніше завдання навчання без вчителя – кластеризацію.

Основним завданням кластеризації є поділ заданих даних на незалежні групи, які називаються кластерами. Дані кластери складаються з різних об'єктів, об'єднаних за певною ознакою, які істотно відрізняють об'єкт, від інших груп

Цілі кластеризації:

- Розуміння даних шляхом виявлення кластерної структури. Таке розбиття вибірки допомагає спростити подальшу обробку даних та прийняття рішень, тому що до кожного кластера в результаті застосовується окремий метод аналізу.
- Стиснення даних. Якщо вихідна вибірка надмірно велика, можна скоротити її, залишивши по одному найбільш типовому представнику від кожного кластера.
- Виявлення новизни (novelty detection). Виділяються нетипові об'єкти, які не вдається приєднати до жодного з кластерів.

РОЗДІЛ 2. ОЦІНКА ЯКОСТІ ПОБУДУВАННЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ

2.1 Критерії оцінювання моделі

Вибір моделі в контексті машинного навчання може мати різні значення, що відповідають різним рівням абстракції.

З одного боку, може зацікавити вибір найкращих гіперпараметрів для вибраного методу машинного навчання. Гіперпараметри - це параметри самого методу навчання, які мають бути за замовчуванням, тобто. до побудови моделі. З іншого боку, параметри моделі - це параметри, що виникають в результаті припасування. Наприклад, у моделі логістичної регресії сила регуляризації (а також тип регуляризації, якщо така є) є гіперпараметром, який повинен бути вказаний до підбору, в той час як коефіцієнти підібраної моделі є параметрами моделі. Пошук правильних гіперпараметрів моделі може мати вирішальне значення для продуктивності моделі на заданих даних.

Також вибір найкращого методу навчання (і відповідні їм «оптимальні» гіперпараметри) може відбуватися з набору відповідних алгоритмів машинного навчання. Візьмемо, наприклад, проблему класифікації: чи дає модель логістичної регресії чи класифікатор випадкових лісів (random forests) найкращу продуктивність цієї задачі.

Рекомендована стратегія вибору моделі залежить від кількості доступних даних. За наявності великої кількості даних можна розділити їх у кілька частин, кожна з яких призначена певної мети. Наприклад, для налаштування гіперпараметра розділити дані на три набори: train/validation/test. Train - використовується для навчання якомога більшої кількості моделей, тому що існують різні комбінації гіперпараметрів моделі. Ці моделі потім оцінюються на Validation set, та модель з найкращими характеристиками на цьому наборі перевірки вибирається як модель-переможець. Потім модель перенавчається на

даних Train set + Validation Set з використанням вибраного набору гіперпараметрів, а ефективність узагальнення оцінюється з використанням набору тестів. Якщо ця помилка узагальнення аналогічна помилці валідації, є підстави вважати, що модель добре працюватиме з майбутніми даними. У результаті модель перевіряється на всьому датасеті (train, validation і test set), перш ніж використовувати її в реальному житті.

Оскільки всі дані мають різну структуру, немає загального правила щодо того, як дані мають бути поділені. Типовий поділ становить, наприклад, 50%/25%/25%. У будь-якому випадку набір перевірки має бути досить великим, щоб виміряти різницю у продуктивності.

Для вибору алгоритму, слідуючи наведеним вище міркуванням, можна використовувати кілька наборів train/validation/set, по три на алгоритм. Оскільки цей метод потребує великих даних, ми обговоримо альтернативний метод нижче.

2.2 Основні причини помилок у прогнозах

При розгляді моделей прогнозування, помилки можуть бути розбиті на кілька основних категорій: помилка через усунення, помилка через дисперсію і "непереборна помилка" (шум). Розуміння цих проблем допоможе діагностувати результати моделей та уникнути неточностей у прогнозах.

Розглянемо останню із названих причин – шум. Шум є фундаментальною концепцією машинного навчання і є у всіх даних. Під даним поняттям мається на увазі спотворення даних, тобто щось, що є стороннім чи хибним стосовно вихідного набору даних. Нижче наведено найбільш поширені приклади спотворення даних: наприклад, є дані зображення, знятого в темний час доби або на обладнання з низькою якістю. Такий датасет не відобразить реальне уявлення у текстових даних можуть бути граматичні та семантичні помилки. Говорячи про аудіо або відео даних під шумом можуть матися на увазі сторонні

звуків, запотівання скла і т.д. Отже, при побудові моделі завжди необхідно враховувати шум.

Іншою важливою складовою помилки є усунення, тобто різниця між очікуваним (середнім) прогнозом моделі та реальним значенням. Зміщення відкидають певні пласти даних або класи, що полегшує припущення, зроблені моделлю для побудови цільової функції. Як правило, лінійні алгоритми мають високий розкид, що робить їх швидкими для вивчення та більш простими для розуміння, але менш гнучкими, що призводить до низької прогнозованої продуктивності за складними завданнями, які не допускають спрощення у своїх рішеннях.

Дисперсія відображає різницю між оцінками у певній точці цільової функції при використанні різних навчальних даних. Модель будується на якихось конкретних даних, тому слід очікувати дисперсію у своїх алгоритмах.

Тепер, скажімо, у нас є деякі дані, які ми використовуємо для навчання наших моделей. Припустимо, що вхідні вектори $\{x_1, x_2, x_n\}$ мають пов'язані з ним результати $\{y_1, y_2, \dots, y_n\}$. Тоді ми можемо побудувати таку формулу будівництва моделі 1.1:

$$y = f(x) + \epsilon \tag{1.1}$$

Мета будь-якого алгоритму - це приведення до мінімуму усунення та дисперсію, тобто "епсілон". Проте існує певна тенденція між алгоритмами: алгоритми лінійного машинного навчання часто мають високий розкид, але низьку дисперсію. Нелінійні алгоритми навпаки часто мають низьке зміщення, але високу дисперсію. Параметризація моделей часто є битвою у пошуку балансу між зміщенням та дисперсією.

Нижче наведено графічну візуалізацію усунення та дисперсії, використовуючи діаграму «bullseye diagram». Центр мети – це модель, яка ідеально передбачає правильні значення. У міру віддалення від центру прогнози стають гіршими. Розглянемо чотири різних випадки, що становлять комбінації як високого, і низького зміщення, і дисперсії.

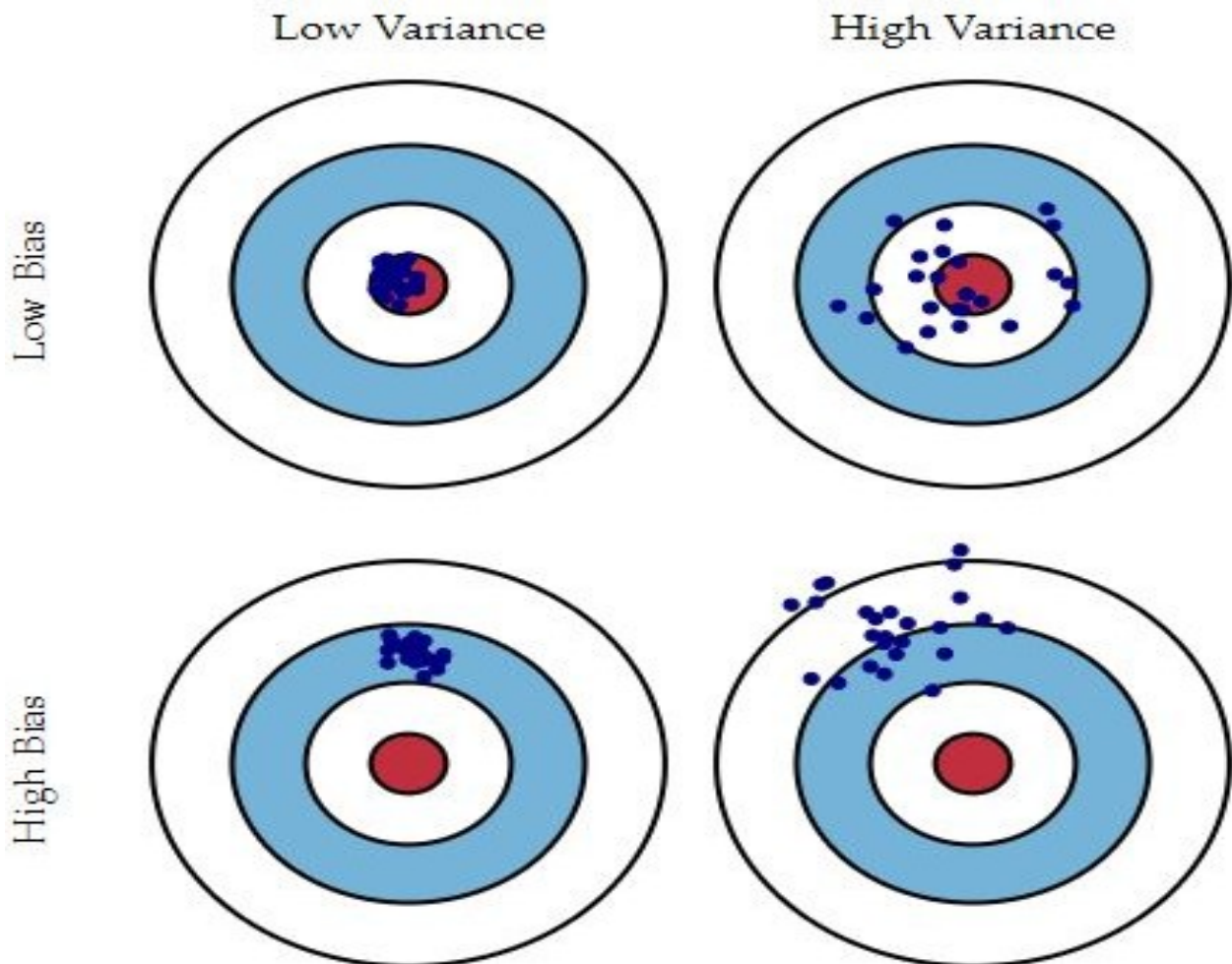


Рисунок 1.4 – Діаграма комбінацій дисперсії та зміщення

Візьмемо псевдо-реальну задачу з простою моделлю та проілюструємо значення усунення та дисперсії при її побудові. Припустимо, ми хочемо побудувати модель для визначення відсотка голосів на виборах за “Республіканців”. Візьмемо вибірку людей із телефонної книги та дізнаємося за кого вони планують голосувати. У результаті отримали результати як на рисунку 1.5:

Voting Republican	Voting Democratic	Non-Respondent	Total
13	16	21	50

Рисунок 1.5 – Приклад розподілення голосів в США

Виходячи з отриманих даних можна зробити висновок, що за “Республіканців” проголосує $13/(13+16)$ або 44% виборців. Проте після підрахунків голосів виявилось, що “Республіканці” перемогли. У чому помилка побудови моделі?

Використання телефонної книги для вибору учасників нашого опитування є упередженим. Досліджуючи лише певні класи людей, спотворює результати таким чином, що якщо ми повторимо наші ітерації при побудові моделі, то, як показано на діаграмі вище (нижня ліва), оцінки відсувають нас від мети, але не призводять до збільшення їхнього розкиду.

З іншого боку, маленька вибірка (50 осіб) – явна ознака дисперсії. При збільшенні розміру вибірки результати були б більш послідовними при повторенні опитування. Результати все ще можуть бути дуже неточними через наші великі джерела упередженості, але дисперсія прогнозів буде зменшена. Однак, наш випадок відображає нижня права діаграма, зазначена вище.

Звичайно, дана модель тривіальна і далека від задач моделювання, але дає уявлення про різницю між дисперсією та усуненням. Крім того, просто збільшення вибірки не дасть потрібних результатів, на практиці існує компроміс, де зменшення одного збільшує інше. Мінімізація загальної помилки моделі потребує ретельного врівноваження цих двох форм помилок.

Повернемося до нашої моделі, визначеної раніше, і опишемо математично наші міркування. Ми можемо оцінити модель $f^{\wedge}(X)$ функції $f(X)$, використовуючи лінійні регресії або інший метод моделювання. У цьому

випадку очікувана квадратна помилка прогнозування в точці x дорівнює:

$$Err(x) = E \left[(Y - \hat{f}(x))^2 \right] \quad (1.2)$$

Потім ця помилка може бути розкладена на компоненти зміщення та дисперсії:

$$Err(x) = \left(E[\hat{f}(x)] - f(x) \right)^2 + E \left[\left(\hat{f}(x) - E[\hat{f}(x)] \right)^2 \right] + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error} \quad (1.3)$$

Графічно це можна зобразити так:

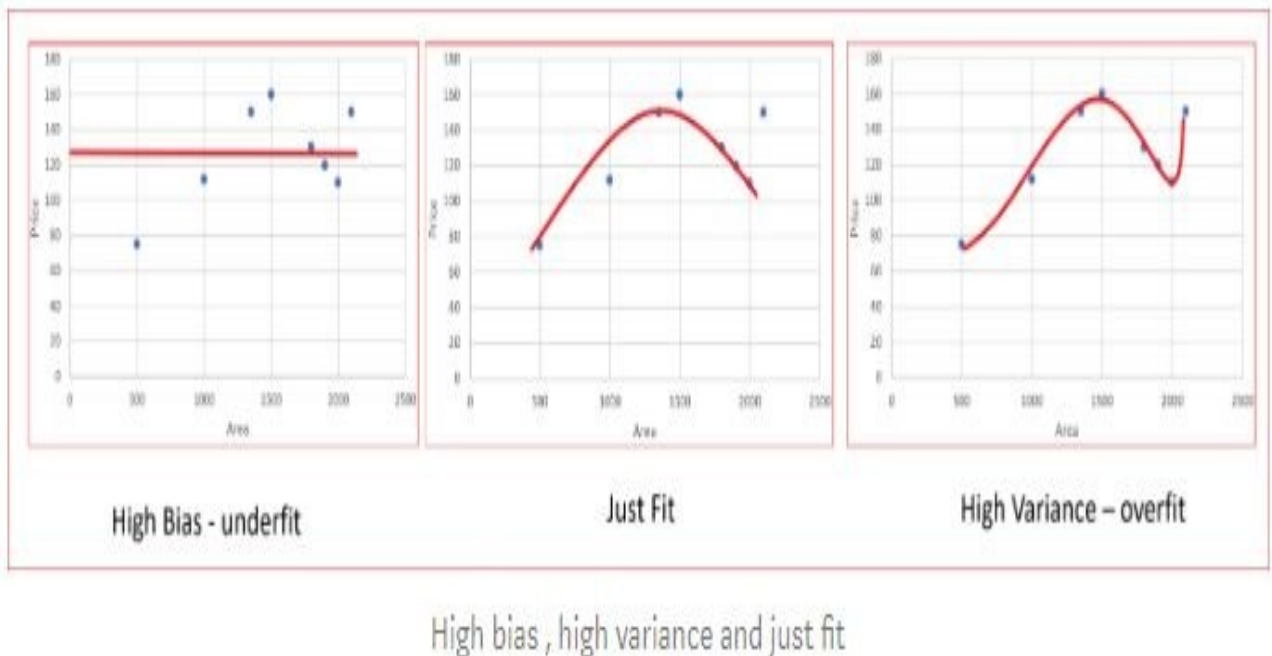


Рисунок 1.5 – Графічне зображення похибки

2.3 Методи валідації моделей

У будь-якому проекті важливо оцінити алгоритм машинного навчання, тому що обрана модель може давати задовільні результати при оцінці з використанням метрики, наприклад `precision_score`, але погані результати в порівнянні з іншими метриками, такими як `logarithmic_loss` або іншими подібними до неї метриками. У більшості випадків використовується Classification Accuracy для вимірювання продуктивності моделі, однак, щоб дійсно оцінити запропоновану модель, це може бути мало. Нижче представлені різні типи метрик та способи оцінок.

Classification Accuracy - у чисельники має обсяг вірних прогнозів стосовно всіх представлених даних.

$$Accuracy = \frac{\textit{Number of Correct predictions}}{\textit{Total number of predictions made}} \quad (1.4)$$

Такий спосіб оцінювання працює ефективно у разі, якщо є однакова кількість даних, що відносяться до кожної групи.

Розглянемо приклад, у якому train set 95% даних групи N і 5% групи M. Таким чином модель покаже точність відповідей 95%, розглядаючи дані, що належать групі N.

Перевіривши цю модель на даних у співвідношення 50% на 50% груп N і M відповідно, точність прогнозу моделі зменшиться і стане 50%. Таким чином, метрика Classification Accuracy може давати хороший результат, але при цьому надавати хибне уявлення.

Мета даної метрики виключати помилкові класифікації, що знаходить ефективне застосування для класифікації з кількома групами. При оцінці Logarithmic Loss дає ймовірнісну оцінку кожної з груп для представлених даних. Розглянемо приклад, у якому N - датасет, що належать до M груп.

$$\text{LogarithmicLoss} = \frac{-1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} * \log(p_{ij}) \quad (1.5)$$

- y_{ij} , означає приналежність даних i до групи j
- p_{ij} , означає ймовірність приналежності даних i до групи j

Даний показник не обмежений зверху та має нижню межу - 0 $[0, \infty)$. Чим ближче оцінка до 0, тим точніше прогноз моделі. У більшості випадків даний показник використовують у задачах класифікації.

Ця матриця служить для оцінки точності моделі. Розглянемо стандартне завдання класифікації, де є дві групи YES та NO. Крім того, дана класифікація з передбачення приналежності групи до представленої датасету. При перегляді результатів моделі на 165 елементів метрика видає наступний результат:

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Таблиця 1.1 – Результат виводу метрики Confusion Matrix

З цього результату можна вивести такі умови:

- кейс, коли модель припустила, що елемент відноситься до класу YES і дійсно до нього відноситься
- випадок, коли прогнозне значення YES, а насправді значення NO

- приклад, коли прогнозне значення NO дійсно відноситься до групи NO
- випадок, коли прогнозне значення NO, а насправді значення YES

Таким чином точність розраховується через формулу 1.6:

$$Accuracy = \frac{TruePositives + FalseNegatives}{TotalNumberofSamples} \quad (1.6)$$

З цієї матриці згодом вивели інші варіації метрик.

Метрика AUC або площа під кривою знаходить широке застосування задач класифікації. Метрика розраховується як ймовірність того, що позитивний приклад буде ранжований вище негативного прикладу. Під визначення AUC лежать такі визначення:

Істинний позитивний коефіцієнт (чутливість): визначається як $TP / (FN+TP)$. Цей показник відповідає частці точок позитивних даних, які правильно вважаються позитивними по відношенню до всіх точок позитивних даних.

$$TruePositiveRate = \frac{TruePositive}{FalseNegative + TruePositive} \quad (1.7)$$

Рівень хибних спрацьовувань (специфічність): визначається як $FP/(FP+TN)$. Невірний позитивний коефіцієнт відповідає частці негативних точок даних, які помилково вважаються позитивними по відношенню до всіх негативних точок даних. Обчислюється через формулу розрахунку від'ємного коефіцієнта метрики AUC.

$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative} \quad (1.8)$$

Неправильний позитивний коефіцієнт та Істинний позитивний коефіцієнт мають значення в діапазоні [0, 1]. FPR та бот TPR обчислюються при порогових значеннях, таких як (0,00, 0,02, 0,04,..., 1,00). Далі будується графік AUC – це

площа під кривою графіка «Невірна позитивна ставка проти істинної позитивної норми» у різних точках [0, 1].

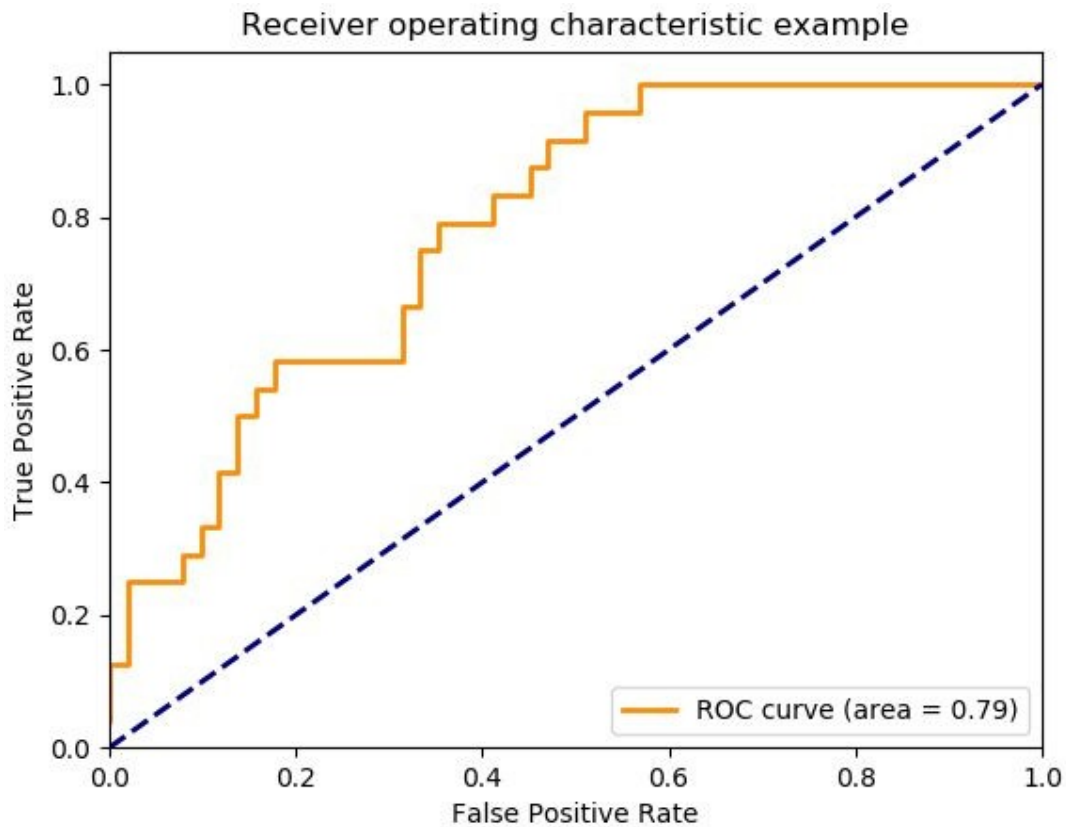


Рисунок 1.6 – Діапазон значень метрики AUC

З цього графіка видно, що показник точності моделі лежить у межах від 0 до 1, причому чим вища оцінка до одиниці, тим ефективніший прогноз.

Оцінка F1 вимірює точність моделі на тестових даних. Даний показник лежить в межах від 0 до 1, показуючи наскільки надійний прогноз моделі, і розраховується за такою формулою.

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

(1.9)

Завданням F1 Score знайти співвідношення між знайти баланс між точністю та відкликанням. Значення лежить у межах від 0 до 1. Показник точності характеризує кількість вірних позитивних відповідей, поділена на кількість позитивних відповідей, і розраховується за формулою.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (2.0)$$

Середня абсолютна помилка показує різницю між реальними даними та результатом прогнозу. Тобто показує відхилення передбаченого значення, але показує напрям помилки. Формула розрахунку наступна 2.1:

$$MeanAbsoluteError = \frac{1}{N} \sum_{j=1}^N |y_j - \hat{y}_j| \quad (2.1)$$

Середня квадратична помилка (MSE) має багато спільного з показником MAE, Різниця полягає в тому, що показник MSE різниця між реальними даними та результатом прогнозу сприймає як середнє значення. Плюсом такої інтерпретації є те, що можна дізнатися градієнт.

$$MeanSquaredError = \frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2 \quad (2.2)$$

2.4 Порівняльний аналіз моделей

По суті всі проблеми машинного навчання є проблемами оптимізації. Завжди є методологія, що лежить в основі моделі машинного навчання, або основна цільова функція, яка має бути оптимізована.

Перевіривши цю модель на даних у співвідношення 50% на 50% груп N і M відповідно, точність прогнозу моделі зменшиться і стане 50%. Таким чином, метрика Classification Accuracy може давати хороший результат, але при цьому надавати хибне уявлення.

Мета даної метрики виключати помилкові класифікації, що знаходить ефективне застосування для класифікації з кількома групами. При оцінці Logarithmic Loss дає ймовірнісну оцінку кожної з груп для представлених даних.

Наприклад, мета моделі лінійної регресії полягає в тому, щоб мінімізувати квадратну втрату прогнозів та фактичне значення (середньоквадратична помилка, MSE), тоді як регресія Лассо спрямована на мінімізацію MSE при обмеженні вивчених параметрів шляхом додавання додаткового члена регулювання.

Насамкінець варто сказати, що алгоритми ML можуть бути проаналізовані на основі різних критеріїв. Ці критерії можуть реально допомогти виміряти ефективність та результативність різних моделей ОД.

РОЗДІЛ 3. РОЗРОБКА КОРИСТУВАЛЬНОГО ІНТЕРФЕЙСУ ДЛЯ АВТОМАТИЗАЦІЇ ПОБУДУВАННЯ ТА ВИКОРИСТАННЯ МОДЕЛІ МАШИННОГО НАВЧАННЯ

3.1. Розробка механізму конфігурації моделей машинного навчання

Аналізовані дані та сфера їх застосування постійно змінюються, зважаючи на різні тренди, дії клієнтів, політичну та економічну ситуації. Вчені постійно спостерігають за різними сплесками в економічному середовищі, які можуть вплинути на прогноз. Крім того, великі компанії постійно працюють над створенням все більш досконалих інструментів для побудови моделей, що вирішують конкретні завдання бізнесу, з чого можна зробити висновок, що на різних даних, та сама модель поводить по-різному. Основною проблемою для замовника є вибір відповідної моделі, а також у постійній її перевірці та актуалізацією аналітиками.

Сьогодні більшість аналітиків визнає, що безперервне навчання моделей (continual learning) є фундаментальним концептом в ML. Це дозволяє в режимі постійного навчання підтримувати новизну моделі щодо сучасних алгоритмів, тенденцій та даних у бойовому середовищі (production).

Analytic Sense пропонує зручне рішення для автоматизації вибору найкращої моделі та безперервного навчання, що підвищує точність моделі, продуктивність та економить час роботи аналітика на перепідготовку, роблячи моделі автоматично адаптованою.

Розглянемо роботу програми з погляду аналітика. Користувачеві надається можливість зберігати моделі та стежити за статусами своїх моделей, доданих раніше. У цьому розділі буде описано роботу з додавання нових рішень. Вибір типу алгоритму складне завдання, крім того, необхідно підібрати ефективні коефіцієнти при конфігурації моделі. Вибір залежить від даних - важливим є розмір вибірки, кількості категоріальних ознак і т.д. також важливо дотриматися

балансу між швидкістю навчання і точністю результату. Аналітик сам обирає алгоритм на вирішення поставленого завдання. Analytic Sense об'єднав найбільш популярні групи алгоритмів, такі як neural network і gradient boosting і засновані на них алгоритми: Catboost, LightGBM, Xgboost і т.д. Згодом можна додавати нові типи алгоритмів, актуалізуючи свою базу, додаючи нові актуальні технології.

3.2. Розробка автоматичного порівняння запропонованих конфігурацій моделей

Далі користувачеві-замовнику необхідно порівняти запропоновані варіанти та вибрати найкращу модель. Analytic Sense автоматично перевіряє моделі на запропонованому датасеті і за допомогою метрик вибирає найефективніший алгоритм. За замовчуванням оцінка проводиться методом МНК, однак користувач може встановити інші способи, наприклад, метрики (AUC, AUCPR і т.д.), які підходять під дані користувача.

Алгоритми було розроблено на вирішення конкретних проблем. Тому важливо знати, з якою проблемою ми маємо справу та який алгоритм найкраще підходить для кожного типу проблем. Ми розглянемо проблеми класифікації та регресії.

Залежно від розміру даних програма автоматично підбирає сімейство алгоритмів, так Байєсовські алгоритми краще працюють на невеликому наборі даних порівняно з алгоритмами Knn, за рахунок низької дисперсії і високим розкидом.

Точність для кожного користувача своя і в залежності від застосування потрібна точність буде відрізнятися. Іноді наближення є виправданим, що може спричинити значне скорочення часу обробки. Крім того, наближені методи дуже стійкі до переоснащення. Також у користувача можуть бути обмежені тимчасові ресурси і т.д. (кількість параметрів та feature)

Ця програма дозволяє на основі вище перерахованих параметрів вибрати відповідне сімейство алгоритмів для подальшої роботи з ними. На основі одержаних результатів від різних моделей за допомогою метрик вибираємо найкращий варіант.

3.3. Безперервне перенавчання побудованої моделі

Після цього постає проблема підтримка точності роботи моделі. Ситуація на ринку постійно змінюється, з'являються нові відомості, тому необхідно оновити модель. Можна щоразу проганяти весь процес навчання наново, оновлюючи свій датасет, перевіряючи параметри моделі тощо. Наш сервіс дозволяє виконувати весь процес навчання автоматично, налаштувавши один раз джерело даних, вимоги до моделі та параметри можна проводити дані операції автоматично. Модель користувача завжди буде навчатися на останніх даних. Якщо налаштування стали старіти, сервіс повідомить користувача і можна вчасно скоригувати параметри.

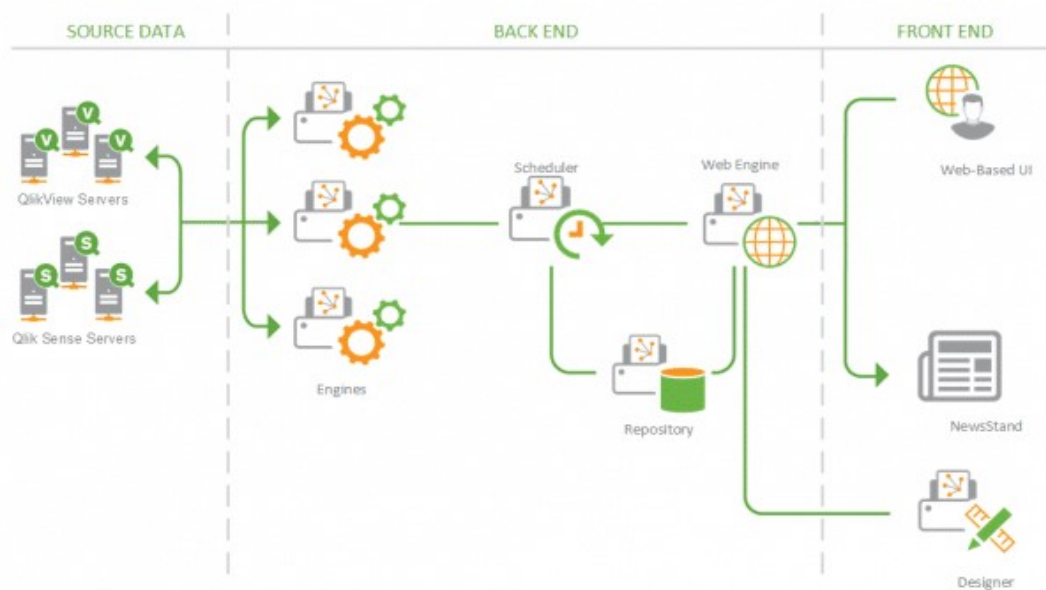


Рисунок 1.7 – Робота застосунку Analytic Sense з точки зору

3.4. Актуальність запропонованого рішення для роботи аналітика та бізнесу

Аналізовані дані та сфера їх застосування постійно змінюються, зважаючи на різні тренди, дії клієнтів, політичну та економічну ситуації. Вчені постійно спостерігають за різними сплесками в економічному середовищі, які можуть вплинути на прогноз. Крім того, великі компанії постійно працюють над створенням все більш досконалих інструментів для побудови моделей, що вирішують конкретні завдання бізнесу, з чого можна зробити висновок, що на різних даних, та сама модель поводить по-різному. Основною проблемою для замовника є вибір відповідної моделі, а також у постійній її перевірці та актуалізацією аналітиками.

Сьогодні більшість аналітиків визнає, що безперервне навчання моделей (continual learning) є фундаментальним концептом в ML. Це дозволяє в режимі постійного навчання підтримувати новизну моделі щодо сучасних алгоритмів, тенденцій та даних у бойовому середовищі (production).

Analytic Sense пропонує зручне рішення для автоматизації вибору найкращої моделі та безперервного навчання, що підвищує точність моделі, продуктивність та економить час роботи аналітика на перепідготовку, роблячи моделі автоматично адаптованою.

Інструмент дозволяє користувачеві додати різні джерела даних (API, SQL таблиці, файли тощо). Бізнес ставить за мету - наприклад, поліпшення результатів скорингового балу для банку, результатом якої буде аналіз платоспроможності клієнта. Також можна додати додаткові критерії оцінювання прогнозу моделей: за замовчуванням цей МНК, однак замовник може встановити інші метрики (AUC, AUCPR тощо).

Після цього користувачі сервісу (аналітики) пропонують моделі для вирішення поставленої мети. Економісти самі вирішують, яким типом алгоритму користуватися і які параметри підібрати. Після проведення аналітичної роботи

користувачі можуть зберегти свої конфігурації моделей з підібраними коефіцієнтами. Analytic Sense перевіряє кожен модель на точність за допомогою встановлених раніше метрик та запускає у production найкращу модель. Проте сервіс зберігає всі запропоновані рішення щодо майбутньому ретроспективи. Після заданого замовником інтервалу часу сервіс автоматично перевіряє поточну модель, а також збережені моделі раніше. Якщо інша модель починає давати результат краще, вона автоматично підміняє поточну модель.

Тепер аналітична робота проходить один раз, вибір моделі та оновлення відбувається автоматично, що дозволяє скоротити час між перемиканням моделей і зберегти бізнес ресурси. Дозволяє автоматично впроваджувати нові моделі та вибирати найкращу.

3.5. Ефективність запропонованого рішення для роботи аналітика та бізнесу

Розглянемо роботу програми з прикладу реального завдання банку Home Credit. У 2019 році організація вирішила переглянути свою політику в оцінюванні кредитоспроможності клієнтів та розробити скорингову модель. Проблема є класичним завданням класифікації, де 0 – не видавати кредит, 1 – видавати. Банк виклав у відкритий доступ дані для побудови моделі. У змаганні брало участь кілька тисяч команд із використанням різних технологій. Запропоновані моделі оцінювалися за допомогою метрики AUC.

Analytical-sense є програмою з веб-інтерфейсом, тобто для взаємодії з сервісом використовується сукупність веб-сторінок.

- Сторінка авторизації та реєстрації

Рисунок 1.8 – Сторінка авторизації та реєстрації

На цій сторінці користувач може зареєструвати свій особистий обліковий запис, який відкриває доступ до іншого функціонала.

Користувач може завантажити список факторів у форматі JSON.

- Сторінка редагування та тестування факторів

Рисунок 1.9 – Сторінка редагування та тестування факторів

У цьому вікні можна перевірити роботу факторів на тестових даних, а також вибрати потрібні фактори для подальшого аналізу.

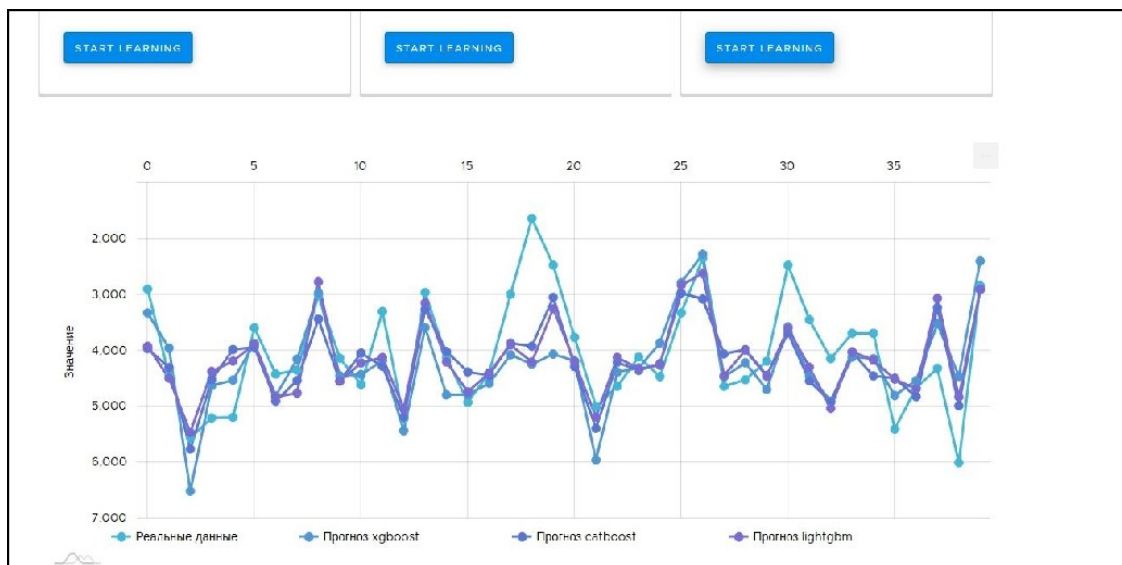


Рисунок 2.0 – Сторінка побудови моделей

Користувач може налаштувати параметри на вибір із трьох алгоритмів: XGBoost, CatBoost, LightGBM; після цього можна запустити навчання на обраних раніше чинниках. Як результат виводиться графік, побудований цими алгоритмами.

На представлених даних найточнішу оцінку дала модель, побудована на LightGBM з точністю 83,8499%. Моделі Xgboost та Catboost показали на кілька відсотків гірші результати. Однак, коли йдеться про мільйони клієнтів та великі кошти, важлива кожна частка. Моделі дали результат точності 78,7648% і 76,705% відповідно. Нижче наведено результати моделей.

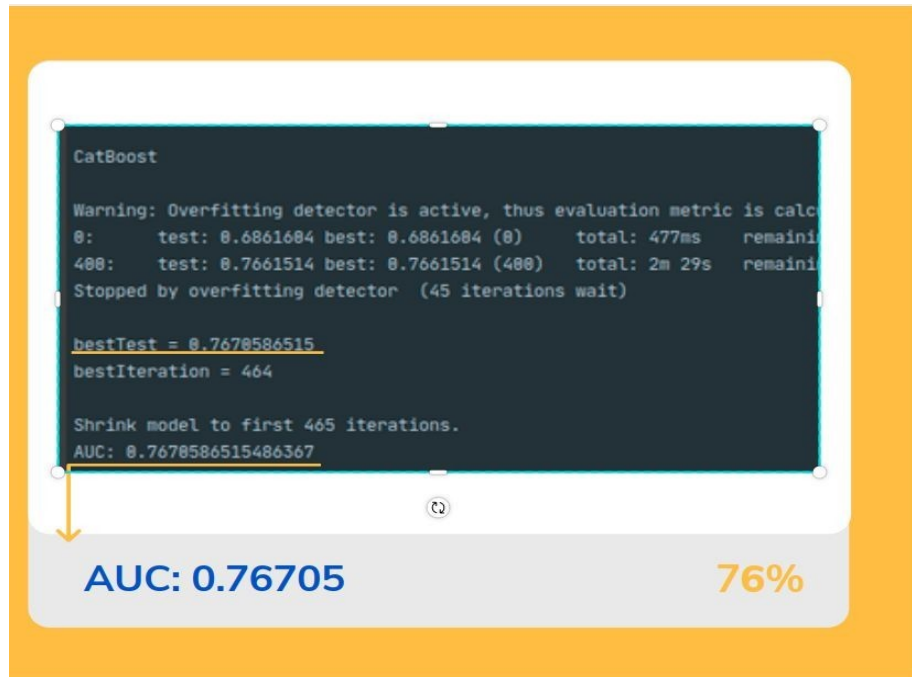


Рисунок 2.1 – Робота Catboost на даних 2022

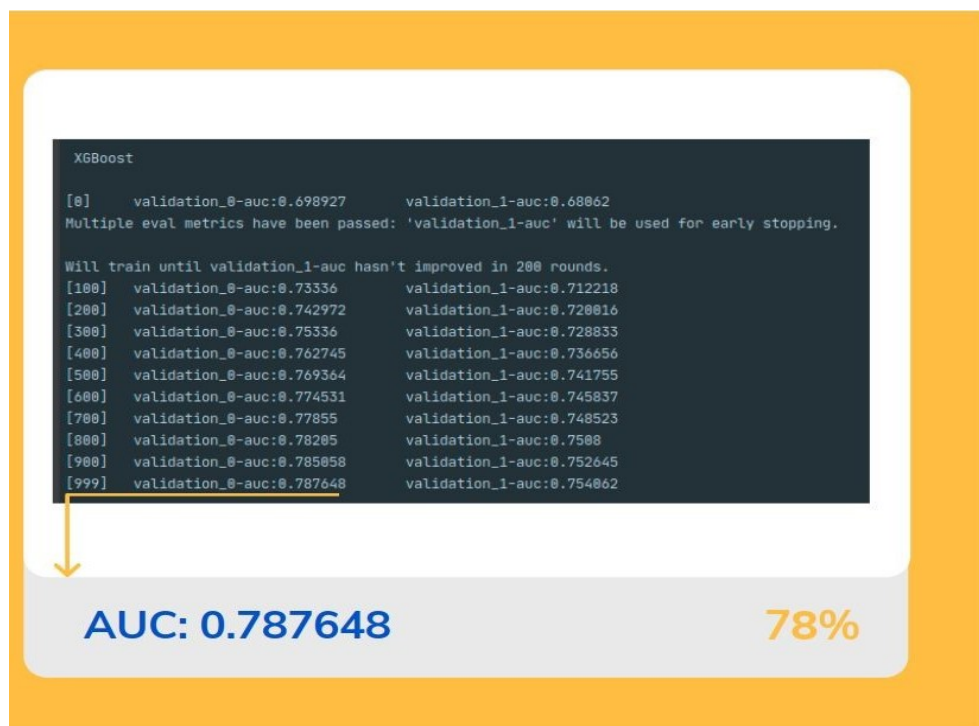


Рисунок 2.2 – Робота Xgboost на даних 2022

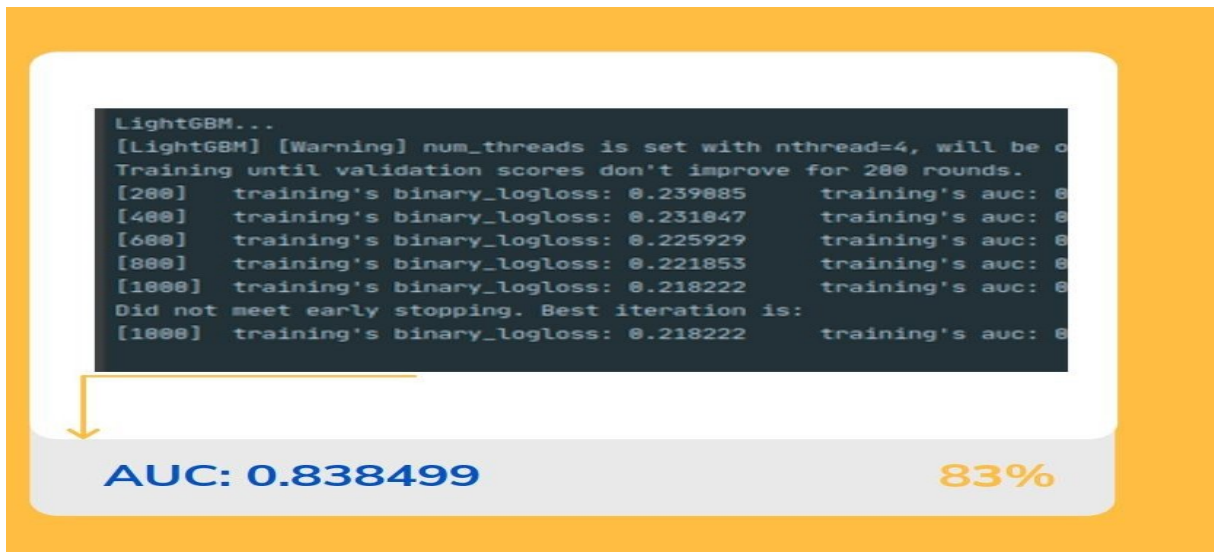


Рисунок 2.3 – Робота LightGBM на даних 2022

Далі, після отримання результатів роботи моделей, програма автоматично вибирає найефективнішу модель, у разі LightGBM.

У 2022 році уз'явилися нові дані, але процес розробки та вибору найкращої моделі займає багато часу та коштів. Тому сервіс запам'ятовує всі запропоновані раніше варіанти від аналітиків, а банку достатньо завантажити нові дані та переглянути результати моделей на актуальних даних. Далі сервіс автоматично вибере найкращу модель і замінить її обрану раніше модель. Розглянемо нижче точність моделей даних 2022 року.

З цих показників можна зробити кілька висновків. По-перше, результати моделей змінилися, і тепер найефективнішою моделлю є XGBoost, що говорить про те, що не існує одного ефективного рішення і для того, щоб показувати ефективний результат, модель необхідно постійно оновлювати та перевіряти її точність. Тому сервіс Analytic Sense пропонує користувачам налаштувати процес навчання та порівняння один раз і далі впроваджувати у виробництво найефективнішу модель автоматично.

Generative Adversarial Networks (GAN) на даний момент є найбільш схожими на людей-митців у світі штучного інтелекту. У цьому типі моделі нейронні мережі виконують дві ролі: Критик (або Дискримінатор) і Генератор. Як пояснювалося раніше, Генератор буде відповідати за виконання підказки та створення зображення. Він дізнається про стиль і зміст, що дозволяє змішувати стилі або інтерполювати між ними. Потім зображення можна створити з нуля. Критик вчиться на вже існуючому мистецтві та образах. Ця база даних знань використовується для порівняння із зображенням, створеним Генератором, щоб визначити, чи може воно виглядати так, ніби його створили люди. Критик намагається виявити фальшиве зображення, яке потім буде надіслано на відтворення. У кожному раунді Генератор вивчає та вдосконалює те, як змусити Критика повірити в реальність мистецтва, що потім може бути використано як кінцевий результат. Цей процес можна повторити з численними підказками, щоб отримати безліч ресурсів для посилення. Таким чином ШІ може навчатися та розвиватися з кожним входом.

Щоб створити зображення, Генератор фактично створює підроблені дані. Мета полягає в тому, щоб передати результат як справжній. Частина GAN, які навчають генератор, включають наступне:

- Зашумлений вхідний вектор
- Мережа генератора (для перетворення випадкового введення в дані)
- Мережа дискримінатора (для класифікації згенерованих даних)
- Втрати генератора (що покарає генератор як форму підкріплення, якщо він не вдало обдурить дискримінатор)

Коли дискримінатор повертає оцінку, зворотне поширення використовується для коригування ваги в правильному напрямку після обчислення впливу ваги на результат. Його також можна використовувати для отримання градієнтів, які можуть змінювати ваги генератора.

Далі докладніше про Дискримінатор. Щоб визначити, справжнє чи підроблене зображення, потрібні тестові дані. Тестові дані надходять із двох різних джерел:

- Зображення реальних даних, які є реальними фотографіями чи картинами, які використовуються як позитивні дані
- Підроблені зображення з Генератора використовуються як негативні дані

Так само, як Генератор карається за те, що не обдурив Дискримінатор, Дискримінатор карається за неправильну класифікацію справжніх і підроблених даних. Він також використовує зворотне поширення для зважування втрат дискримінатора в мережі дискримінатора.

Коли обидва навчені, починається змагальна гра. Генератор намагається обдурити Дискримінатор, стаючи кращим із кожною спробою, а Дискримінатор класифікує справжні чи підроблені дані, також покращуючи кожную неправильно класифіковану точку даних.

Хоча є багато іншого, про що ми можемо дізнатися, давайте швидко розглянемо кроки для навчання GAN, перш ніж рухатися далі:

Як і в інших методах машинного навчання, машина навчається сама з мінімальним втручанням людей. Це означає, що коли ви бачите кінцевий результат згенерованих зображень, це, ймовірно, не єдине групування зображень. Ймовірно, є багато інших, які ви не бачите, оскільки вони не набрали достатньо високих балів, щоб Генератор міг їх повернути. Що стосується обмежень, то машина тренується сама. Це означає, що дані не завжди точні. Комп'ютер не розуміє зображення, які він переглядає, оскільки він не може «бачити» зображення так, як ми можемо. Він також не може розрізнити, що він «повинен» або «не повинен» створювати. Тому є обмеження, але технологія розвивається.

Однак одним обмеженням є ваша творчість. Оскільки машина не знає, що вона повинна або не повинна створювати, вона намагатиметься зробити все, що ви їй дасте. Отже, будь-яка підказка, яку ви можете придумати. Тепер вони не завжди виглядають «добре», як у випадках з Dall-E. З Dall-E деякі зображення справді

неймовірні. Але інші можуть виглядати трохи жахливо. Це все в процесі навчання. З Midjourney майже не так багато зображень на основі «людей». Пейзажі, які він може створювати, дійсно прекрасні, але зображення більш «стилістичні», більше схожі на картини, ніж на фотографії.

Останнє питання, яке мене цікавило, це те, як штучний інтелект може отримати таку величезну кількість даних. Але відповідь не зовсім зрозуміла. Деякі штучні інтелекти мають доступ лише до того, що їм надають інформатики. Інші штучні інтелекти використовують веб-збирання, щоб мати доступ до мільярдів фотографій для довідки. Підписи також знімаються та використовуються в процесі навчання. Зображення ідентифікуються за підписами, і тому він дізнається, які підписи мають поєднуватися з певними зображеннями. Таким чином, введення тексту, яке буде підписом, може бути згенеровано на основі того, яке зображення «має» поєднуватися з підписом.

Зараз вчені Google винайшли дифузійні обчислювальні моделі, які сьогодні є основою генераторів зображень, але компанія настільки стурбована тим, що люди можуть з ними робити, що досі не відкрила власні експериментальні генератори Imagen і Parti для громадськості. (Лише співробітники можуть спробувати їх, дотримуючись суворих вказівок щодо того, що можна вимагати.) Тому не випадково, що три найпопулярніші платформи для генераторів зображень наразі — це три стартапи, які не мають спадщини, яку потрібно захистити. Midjourney — це початковий стартап, запущений Девідом Хольцом, який заснував генератор у новій спільноті художників. Інтерфейсом для ШІ є галасливий сервер Discord; вся робота та підказки були оприлюднені з самого початку. DALL-E — це продукт другого покоління некомерційної організації OpenAI, що фінансується Ілоном Маском та іншими. Stable Diffusion з'явився на сцені в серпні 2022 року, створений європейським підприємцем Емадом Мостакє. Це проєкт із відкритим вихідним кодом, додаткова перевага якого полягає в тому, що кожен може завантажити його програмне забезпечення та запустити його локально на

власному комп'ютері. Більше ніж інші, Stable Diffusion випустила генератори зображень AI у дику природу.

Чому так багато людей з таким захопленням грають із цим ШІ? Багато зображень створюються з тієї ж причини, з якої люди завжди створювали більшість мистецтва: тому що зображення красиві, і ми хочемо на них дивитися. Як полум'я у багатті, світлові візерунки заворожують. Вони ніколи не повторюються; вони дивують знову і знову. Вони зображують сцени, свідками яких ніхто раніше не бачив і навіть не міг собі уявити, і вони майстерно складені. Це таке ж задоволення, як досліджувати світ відеоігор або гортати художню книгу. У їхній творчості справжня краса, і ми дуже дивимося на те, як ми могли б оцінити чудову художню виставку в музеї. Насправді перегляд парадусгенерованих зображень дуже схожий на відвідування особистого музею, але в цьому випадку стіни сповнені мистецтва, яке ми хочемо. І вічна новизна та несподіванка чергового образу майже не слабшає. Користувачі можуть ділитися знайденими дорогоцінними каменями, але я припускаю, що 99 відсотків із 20 мільйонів зображень, які зараз генеруються щодня, зможе переглянути лише одна людина — їхній співавтор.

Як і будь-яке мистецтво, образи також можуть бути цілющими. Люди витрачають час на створення дивних зображень штучного інтелекту з тієї самої причини, з якої вони можуть малювати в неділю, чи писати в щоденнику, чи знімати відео. Вони використовують засоби масової інформації, щоб визначити щось у своєму власному житті, те, про що інакше сказати не можна. Я бачив зображення, які зображують, як може виглядати рай для тварин, створений у відповідь на смерть улюбленого собаки. Багато зображень досліджують представлення нематеріальних, духовних сфер, мабуть, як спосіб думати про них. «Величезна частина всього використання — це в основному арт-терапія», — каже мені Хольц, творець Midjourney. «Зображення насправді не є естетично привабливими в універсальному сенсі, але привабливі в дуже глибокому сенсі в контексті того, що відбувається в житті людей». Машини можна використовувати

для створення фантазій усіх типів. Хоча розміщені служби забороняють порно та кровопролитну сцену, у версіях для настільних комп'ютерів можна все, як це можливо у Photoshop.

Зображення, створені ШІ, також можуть бути корисними. Скажімо, ви презентуєте доповідь про можливість переробки лікарняних пластикових відходів у будівельні матеріали та хочете зображення будинку, зробленого з пробірок. Ви можете шукати на фондових ринках фотографій придатне для використання зображення, створене людиною-митцем. Але унікальне призначення, подібне до цього, рідко дає вже існуюче зображення, і навіть якщо його буде знайдено, його статус авторського права може бути сумнівним або дорогим. Дешевше, швидше і, ймовірно, набагато доцільніше створити унікальне, персоналізоване зображення для вашого звіту за кілька хвилин, яке потім можна вставити у ваші слайди, інформаційний бюлетень або блог, і право власності на авторські права (на даний момент) належить вам. Я сам використовую ці генератори для спільного створення зображень для власних слайд-презентацій.

Під час неофіційного опитування досвідчених користувачів я виявив, що лише близько 40 відсотків свого часу витрачають на пошук утилітарних зображень. Більшість зображень ШІ використовуються в місцях, де раніше не було зображень. Зазвичай вони не замінюють зображення, створене людиною-митцем. Вони можуть бути створені, наприклад, для ілюстрації лише текстового бюлетеня

Сьогодні у випадку з генераторами зображень зі штучним інтелектом нова група художників і фотографів, які дуже підковані в техніці, працює над панікою рівня 3. У відповідь, гіпотетично від третьої особи, вони бояться, що інші люди (але ніколи вони самі) можуть втратити роботу. Getty Images, провідне агентство з продажу стокових фотографій та ілюстрацій для дизайну та редакційного використання, вже заборонило зображення, створені ШІ; деякі художники, які розміщують свої роботи на DeviantArt, вимагали подібної заборони. Існують вимоги з благими намірами ідентифікувати мистецтво штучного інтелекту ярликом і відокремити його від «справжнього» мистецтва.

Крім того, деякі митці хочуть отримати гарантії, що їхні власні роботи не будуть використовуватися для навчання ШІ. Але це типово для паніки 3-го рівня — у кращому випадку вона є помилковою. Алгоритми піддаються 6 мільярдам зображень із супутнім текстом. Якщо ви не є впливовим художником, видалення вашої роботи не має жодного значення. Згенероване зображення виглядатиме абсолютно однаково з вашою роботою в навчальному наборі чи без неї. Але навіть якщо ви впливовий художник, видалення ваших зображень все одно не матиме значення. Оскільки ваш стиль вплинув на роботу інших — визначення впливу — ваш вплив залишиться, навіть якщо ваші зображення буде видалено. Уявіть собі, якби ми прибрали всі картини Ван Гога з навчального набору. Стиль Ван Гога все одно залишився б у величезному океані образів, створених тими, хто його наслідував або перебував під його впливом.

Стилі викликаються за допомогою підказок, наприклад: «у стилі Ван Гога». Деякі нещасні артисти хотіли б, щоб їхні імена були піддані цензурі та заборонені використовувати їх як підказку. Таким чином, навіть якщо їхній вплив неможливо усунути, ви не можете досягти його, оскільки їх ім'я заборонено. Як ми знаємо з усіх попередніх спроб цензури, такі заборони на слова легко обійти; ви можете написати назву з помилкою або просто описати стиль словами. Наприклад, я виявив, що можу створювати детальні чорно-білі фотографії природного ландшафту з величезним освітленням і помітним переднім планом, навіть не використовуючи ім'я Анселя Адамса.

Є ще одна мотивація для художника зніматися. Вони можуть боятися, що велика корпорація зароблятиме гроші на їхній роботі, а їхній внесок не буде компенсований. Але ми не компенсуємо митцям за їхній вплив на інших митців. Візьмемо Девіда Хокні, одного з найбільш високооплачуваних нині живих художників. Хокні часто визнає великий вплив інших нинішніх художників на його творчість. Як суспільство, ми не очікуємо, що він (або інші) будуть виписувати чеки на свій вплив, хоча він міг би. Вважати, що штучний інтелект

повинен платити своїм впливовим особам, це натяжка. «Податок», який успішні митці сплачують за свій успіх, — це їхній неоплачений вплив на успіх інших.

Більше того, лінії впливу є розмитими, ефемерними та неточними. На всіх нас впливає все, що нас оточує, до певної міри, про що ми не усвідомлюємо і, звичайно, не можемо визначити кількісно. Коли ми пишемо пам'ятку чи робимо фотографію за допомогою телефону, наскільки на нас вплинули — прямо чи опосередковано — Ернест Хемінгуей чи Доротея Ланге? Неможливо розгадати наш вплив, коли ми щось створюємо. Так само неможливо розгадати нитки впливу у всесвіті зображень ШІ. Теоретично ми могли б побудувати систему для виплати грошей, зароблених штучним інтелектом, виконавцям у навчальному наборі, але ми повинні визнати, що цей кредит буде надано доволіно (несправедливо) і що фактичні компенсаційні суми на виконавця в групі з 6 мільярд акцій було б настільки тривіальним, щоб бути безглуздим.

У найближчі роки обчислювальний механізм у генераторі зображень ШІ продовжуватиме розширюватися та вдосконалюватися, поки не стане центральним вузлом у всьому, що ми робимо візуально. Воно бачило буквально все і знало всі стилі, малюватиме, уявлятиме та створюватиме практично все, що нам потрібно. Він стане візуальною пошуковою системою, візуальною енциклопедією, за допомогою якої можна розуміти зображення, і основним інструментом, який ми використовуємо з нашими найважливішими відчуттями, зором. Наразі кожен алгоритм нейронної мережі, що працює глибоко в ШІ, покладається на величезні обсяги даних, отже, мільярди зображень, необхідні для його навчання. Але в наступне десятиліття ми матимемо оперативний штучний інтелект, який покладається на набагато менше прикладів для навчання, можливо, лише на 10 000. Ми навчимо ще більш потужні генератори зображень штучного інтелекту, як малювати, показавши їм тисячі ретельно підібраних, ретельно відібраних зображень існуючого мистецтва, і коли цей момент настане, митці з різним походженням будуть боротися один з одним, щоб потрапити в навчальний набір. . Якщо митець входить до основного пулу, його вплив поділять і відчують

усі, тоді як ті, хто не включений, повинні подолати головну перешкоду для будь-якого митця: не піратство, а невідомість.

Що мистецтво, створене AIGenerated, насправді означає для людської творчості. Щойно народилися двовимірні генеративні алгоритми, експериментатори кинулися з'ясовувати, що буде далі. Дженсен Хуанг, амбітний співзасновник Nvidia, вважає, що наступне покоління чіпів створить 3D-світи для метавсесвіту — «наступної обчислювальної платформи», як він її називає. Протягом одного тижня у вересні минулого року було оголошено три нові генератори тексту в 3D/відео: GET3D (Nvidia), Make-A-Video (Meta) і DreamFusion (Google). Розширення відбувається швидше, ніж я можу написати. Якими б дивовижними не були створювані штучним інтелектом кадрові 2D-зображення, аутсорсинг їх створення не змінить радикально світ. Ми вже на піку 2D. Генератори зображень зі штучним інтелектом створюватимуть 3D-зображення та відео.

Майбутня підказка для 3D-механізму може виглядати приблизно так: «Створіть безладну спальню підлітка з плакатами на стіні, незастеленим ліжком і сонячним світлом, що пробивається крізь закриті жалюзі». І за лічені секунди народжується повністю реалізована кімната, відкриті дверцята шафи та весь брудний одяг на підлозі — у повному 3D. Потім скажіть ШІ: «Зробіть кухню 1970-х з магнітами на холодильник і всіма коробками з крупами в коморі. У повній об'ємній деталізації. Такий, через який можна пройти. Або це можна було б сфотографувати на відео». Ігри, наповнені альтернативно відтвореними світами, і повнометражні фільми, прикрашені костюмами та декораціями, завжди були недосяжними для окремих художників, які залишаються під владою великих доларів. ШІ міг створювати ігри, метавсесвіти та фільми так само швидко, як романи, картини та пісні. Фільми Pixar миттєво! Коли мільйони аматорів створять мільярди фільмів і нескінченні метавсесвіти вдома, вони виведуть абсолютно нові жанри медіа — віртуальний туризм, просторові меми — за допомогою своїх

рідних геніїв. І коли значні кошти та професіонали отримають ці нові інструменти, ми побачимо шедеври такого рівня складності, якого раніше не бачили.

Але навіть величезні всесвіти 3D-світів і відео недостатньо великі, щоб стримати зрив, спричинений генераторами зображень ШІ. DALL-E, Midjourney і Stable Diffusion — лише перші версії генеративних машин усіх типів. Їх основна функція, розпізнавання образів, є майже рефлексом для людського мозку, те, що ми виконуємо без свідомого мислення. Це в основі майже всього, що ми робимо.

Замість того, щоб боятися ШІ, нам краще подумати про те, чого він нас навчає. І найголовніше, чого вчать нас генератори зображень штучного інтелекту: творчість — це не якась надприродна сила. Це те, що можна синтезувати, посилювати та маніпулювати. Виявляється, нам не потрібно було досягти інтелекту, щоб вилупити креативність. Творчість більш елементарна, ніж ми думали. Це не залежить від свідомості. Ми можемо створити креативність у чомусь такому дурному, як нейронна мережа глибокого навчання. Величезних даних і алгоритмів розпізнавання образів здається достатніми для розробки процесу, який дивуватиме і допомагатиме нам безперервно.

Дослідники креативності називають те, що називається креативністю у верхньому регістрі. Uppercase Creativity — це приголомшливе перевлаштування, що змінює поле та світ, яке приносить великий прорив. Згадайте спеціальну теорію відносності, відкриття ДНК або Герніку Пікассо. Творчість у верхньому регістрі виходить за рамки просто нового. Він особливий і рідкісний. Це глибоко зворушує нас, людей, далеко за межі того, що може досягнути інопланетний штучний інтелект.

Для глибокого зв'язку з людиною завжди потрібна творча людина в курсі. Однак цю високу креативність не слід плутати з креативністю, яку більшість художників, дизайнерів і винахідників виробляють щодня. Буденна, звичайна креативність у нижньому регістрі — це те, що ми отримуємо з чудовим новим дизайном логотипу чи крутою обкладинкою книги, вишуканим цифровим носієм чи останньою модою, що має бути обов'язковою, чи дизайном сцен для нашого

улюбленого науково-фантастичного серіалу. Більшість людського мистецтва, минулого та теперішнього, написані малими літерами. І креативність у нижньому регістрі – це саме те, що забезпечують генератори ШІ.

Але це величезне. Вперше в історії люди можуть створювати повсякденні акти творчості за запитом, у реальному часі, у великих масштабах, недорого. Синтетична творчість зараз є товаром. Стародавні філософи перевернуться в могилах, але виявляється, що для розвитку творчості — для створення чогось нового — все, що вам потрібно, це правильний код. Ми можемо вставити його в крихітні пристрої, які зараз інертні, або ми можемо застосувати креативність до великих статистичних моделей, або вставити креативність у процедури відкриття ліків. Для чого ще можна використовувати синтетичну творчість? Ми можемо відчувати себе трохи схожими на середньовічних селян, яких запитують: «Що б ви зробили, якби у вас під рукою була сила 250 коней?» Ми не знаємо. Це надзвичайний подарунок. Що ми знаємо, так це те, що тепер у нас є легкі двигуни творчості, які ми можемо націлити в застарілі кутки, де ніколи не бачили новизни, інновацій або вау творчих змін. На фоні всього, що ламається, ця суперсила може допомогти нам продовжити вау на невизначений термін. При належному застосуванні ми можемо зробити невелику вм'ятину у Всесвіті.

3.6 Генеративний ШІ

Генеративний ШІ вже може зробити багато. Він здатний створювати текст і зображення, охоплюючи дописи в блогах, програмний код, поезію та твори мистецтва (і навіть перемагати в конкурсах, що суперечливо). Програмне забезпечення використовує складні моделі машинного навчання, щоб передбачити наступне слово на основі попередніх слів або наступне зображення на основі слів, що описують попередні зображення. LLM почали працювати в Google Brain у 2017 році, де спочатку вони використовувалися для перекладу слів із збереженням контексту. Відтоді великі мовні моделі та моделі перетворення

тексту в зображення поширилися у провідних технологічних фірмах, зокрема Google (BERT і LaMDA), Facebook (OPT-175B, BlenderBot) і OpenAI, неприбутковій організації, в якій Microsoft є домінуючим інвестором (GPT-3 для тексту, DALL-E2 для зображень і Whisper для мовлення). Такі онлайн-спільноти, як Midjourney (яка допомогла виграти мистецький конкурс), і постачальники з відкритим кодом, такі як HuggingFace, також створили генеративні моделі.

Ці моделі здебільшого обмежуються великими технологічними компаніями, оскільки їх навчання потребує величезних обсягів даних і обчислювальної потужності. GPT-3, наприклад, спочатку навчався на 45 терабайтах даних і використовує 175 мільярдів параметрів або коефіцієнтів для прогнозування; один навчальний запуск для GPT-3 коштував 12 мільйонів доларів. Wu Dao 2.0, китайська модель, має 1,75 трильйона параметрів. Більшість компаній не мають можливостей центру обробки даних або бюджету хмарних обчислень, щоб навчати власні моделі такого типу з нуля.

Але після того, як генеративну модель навчено, її можна «точно налаштувати» для певної області вмісту з набагато меншою кількістю даних. Це призвело до створення спеціалізованих моделей BERT — для біомедичного вмісту (BioBERT), юридичного вмісту (Legal-BERT) і тексту французькою мовою (CamemBERT) — і GPT-3 для різноманітних конкретних цілей. BioNeMo від NVIDIA — це платформа для навчання, побудови та розгортання великих мовних моделей у суперкомп'ютерному масштабі для генеративної хімії, протеоміки та ДНК/РНК. OpenAI виявив, що лише 100 конкретних прикладів предметно-спеціальних даних можуть значно підвищити точність і релевантність виходів GPT-3.

Для ефективного використання генеративного штучного інтелекту все ще потрібна людська участь як на початку, так і в кінці процесу.

Почнемо з того, що людина повинна ввести підказку в генеративну модель, щоб вона створювала вміст. Загалом, творчі підказки дають творчі результати. «Інженер оперативного управління», ймовірно, стане усталеною професією,

принаймні до появи наступного покоління ще розумнішого ШІ. Ця сфера вже призвела до створення 82-сторінкової книги підказок із зображеннями DALL-E 2 і ринку підказок, де за невелику плату можна купити підказки інших користувачів. Більшості користувачів цих систем доведеться спробувати кілька різних підказок, перш ніж досягти бажаного результату.

Потім, коли модель генерує вміст, людині потрібно буде його ретельно оцінити та відредагувати. Альтернативні підказки можна об'єднати в один документ. Створення зображення може вимагати значних маніпуляцій. Джейсон Аллен, який виграв конкурс «цифрової обробленої фотографії» в Колорадо за допомогою Midjourney, розповів репортеру, що він витратив понад 80 годин, створюючи понад 900 версій мистецтва, і постійно налаштовував свої підказки. Потім він покращив результат за допомогою Adobe Photoshop, підвищив якість і різкість зображення за допомогою іншого інструменту штучного інтелекту та надрукував три частини на полотні.

Генеративні моделі ШІ неймовірно різноманітні. Вони можуть отримувати такий вміст, як зображення, довші текстові формати, електронні листи, вміст соціальних мереж, голосові записи, програмний код і структуровані дані. Вони можуть виводити новий вміст, переклади, відповіді на запитання, аналіз настроїв, резюме та навіть відео. Ці машини для універсального вмісту мають багато потенційних застосувань у бізнесі, деякі з яких ми описуємо нижче.

Ці генеративні моделі є потенційно цінними для низки бізнес-функцій, але маркетингові програми є, мабуть, найпоширенішими. Jasper, наприклад, маркетингова версія GPT-3, може створювати блоги, публікації в соціальних мережах, веб-версію, рекламні електронні листи, рекламу та інші типи контенту для клієнтів. Він стверджує, що часто перевіряє свої результати за допомогою тестування A/B і що його вміст оптимізовано для розміщення в пошукових системах. Jasper також налаштовує моделі GPT-3 на найкращі результати своїх клієнтів, що, за словами керівників Jasper, призвело до суттєвих покращень. Більшість клієнтів Jasper – це окремі особи та малі підприємства, але деякі групи

великих компаній також використовують його можливості. У компанії хмарних обчислень VMWare, наприклад, автори використовують Jasper для створення оригінального контенту для маркетингу, від електронної пошти до кампаній продуктів і копій у соціальних мережах. Роза Лір, директор із розвитку продукту, сказала, що Jasper допоміг компанії розширити нашу контент-стратегію, а автори тепер мають час для кращих досліджень, ідей і стратегій.

Кріс Рубі, власник агентства зі зв'язків з громадськістю та соціальних мереж Ruby Media Group, тепер використовує генерація тексту та зображення з генеративних моделей. Вона каже, що вони ефективні в максимізації пошукової оптимізації (SEO) і в PR, для персоналізованих виступів для авторів. Ці нові інструменти, на її переконання, відкривають новий рубіж у боротьбі з авторським правом, і вона допомагає створювати політику штучного інтелекту для своїх клієнтів. Коли вона користується інструментами, вона каже: «ШІ — це 10%, я — 90%», тому що потрібно багато підказок, редагування та ітерацій. Вона вважає, що ці інструменти роблять написання кращим і повнішим для пошукових систем, і що інструменти для створення зображень можуть замінити ринок стокових фотографій і призвести до ренесансу творчої роботи.

DALL-E 2 та інші засоби генерації зображень вже використовуються для реклами. Хайнц, наприклад, використав зображення пляшки кетчупу з етикеткою, схожою на етикетку Хайнца, щоб стверджувати, що «ось як «кетчуп» виглядає для ШІ». Звичайно, це означало лише те, що модель тренувалася на відносно великій кількості фотографій пляшок кетчупу Heinz. Компанія Nestle використала розширену версію картини Вермеєра за допомогою штучного інтелекту, щоб допомогти продати одну зі своїх марок йогурту. Stitch Fix, компанія з виробництва одягу, яка вже використовує штучний інтелект, щоб рекомендувати клієнтам певний одяг, експериментує з DALL-E 2 для створення візуалізацій одягу на основі запитуваних уподобань клієнтів щодо кольору, тканини та стилю. Mattel використовує цю технологію для створення зображень для дизайну та маркетингу іграшок.

Зокрема GPT-3 виявився ефективним, якщо не ідеальним, генератором комп'ютерного програмного коду. З огляду на опис «фрагмента» або невеликої функції програми, програма GPT-3 Codex — спеціально навчена для створення коду — може створювати код різними мовами. Github від Microsoft також має версію GPT-3 для генерації коду під назвою CoPilot. Найновіші версії Codex тепер можуть виявляти помилки та виправляти помилки у власному коді — і навіть пояснювати, що код робить — принаймні деякий час. Оголошена мета Microsoft полягає не в тому, щоб усунути людей-програмістів, а в тому, щоб зробити такі інструменти, як Codex або CoPilot, «парними програмістами» з людьми, щоб підвищити їх швидкість і ефективність.

Консенсус щодо генерації коду на основі LLM полягає в тому, що він добре працює для таких фрагментів, хоча інтеграція їх у більшу програму та інтеграція програми в конкретне технічне середовище все ще вимагає людських можливостей програмування. Протягом останніх кількох місяців Deloitte активно експериментував із Codex і виявив, що він підвищує продуктивність для досвідчених розробників і створює деякі можливості програмування для тих, хто не має досвіду.

У шеститижневому пілотному проекті Deloitte з 55 розробниками протягом 6 тижнів більшість користувачів оцінили точність отриманого коду на рівні 65% або вище, причому більшість коду надійшло від Codex. Загалом експеримент Deloitte виявив 20% покращення швидкості розробки коду для відповідних проектів. Deloitte також використовував Codex для перекладу коду з однієї мови на іншу. Висновок фірми полягав у тому, що в осяжному майбутньому їй все одно знадобляться професійні розробники, але підвищення продуктивності може призвести до того, що їх буде менше. Як і з іншими типами генеративних інструментів ШІ, вони виявили, що чим краще підказка, тим кращий вихідний код.

LLM все частіше використовуються в основі розмовного ШІ або чат-ботів. Вони потенційно пропонують більший рівень розуміння розмови та усвідомлення контексту, ніж поточні розмовні технології. Наприклад, BlenderBot від Facebook,

розроблений для діалогу, може вести довгі розмови з людьми, зберігаючи контекст. Google BERT використовується для розуміння пошукових запитів, а також є компонентом чат-бота DialogFlow компанії. Google LaMBA, ще один магістр права, також був розроблений для діалогу, і розмови з ним переконали одного з інженерів компанії, що це розумна істота — вражаюче досягнення, враховуючи, що він просто передбачає слова, які використовуються в розмові, на основі минулих розмов.

Жоден із цих LLM не є ідеальним співрозмовником. Вони навчаються на основі минулого людського контенту та мають тенденцію повторювати будь-яку расистську, сексистську чи упереджену мову, з якою вони стикалися під час навчання. Хоча компанії, які створили ці системи, працюють над фільтрацією мови ненависті, вони ще не досягли повного успіху.

Одним із нових застосувань LLM є використання їх як засобу керування текстовими (або, можливо, зображеннями чи відео) знаннями в організації. Трудомісткість створення структурованих баз знань ускладнила масштабне управління знаннями для багатьох великих компаній. Проте деякі дослідження показали, що магістратури можуть бути ефективними в управлінні знаннями організації, коли модельне навчання точно налаштовано на певний обсяг текстових знань в організації. Знання в LLM можуть бути доступні за запитаннями, виданими як підказки.

Деякі компанії досліджують ідею управління знаннями на основі LLM разом із провідними постачальниками комерційних LLM. Наприклад, Morgan Stanley працює з OpenAI GPT-3, щоб налаштувати навчання щодо контенту управління капіталом, щоб фінансові консультанти могли як шукати наявні знання всередині фірми, так і легко створювати індивідуальний контент для клієнтів. Здається ймовірним, що користувачам таких систем знадобиться навчання або допомога у створенні ефективних підказок, і що результати знань LLM все ще потребують редагування або перегляду перед застосуванням. Однак якщо припустити, що такі проблеми будуть вирішені, магістерські програми

можуть відновити сферу управління знаннями та дозволити їй набагато ефективніше масштабуватися.

Ми вже бачили, що ці генеративні системи штучного інтелекту швидко призводять до низки юридичних та етичних проблем. «Дипфейки» або зображення та відео, створені штучним інтелектом і нібито реалістичні, але такими не є, уже з'явилися в засобах масової інформації, розвагах і політиці. Однак раніше створення дипфейків вимагало значних обчислювальних навичок. Тепер, однак, створити їх зможе практично кожен. OpenAI намагався контролювати фальшиві зображення, наносячи «водяний знак» на кожне зображення DALL-E 2 характерним символом. Проте в майбутньому, ймовірно, знадобиться більше засобів контролю, особливо коли генеративне створення відео стане мейнстрімом.

Генеративний штучний інтелект також викликає численні запитання про те, що таке оригінальний і запатентований контент. Оскільки створений текст і зображення не зовсім схожі на будь-який попередній вміст, провайдери цих систем стверджують, що вони належать їх оперативним творцям. Але вони явно є похідними від попереднього тексту та зображень, використаних для навчання моделей. Зайве говорити, що ці технології забезпечать значну роботу для адвокатів з питань інтелектуальної власності в найближчі роки.

З цих кількох прикладів бізнес-додатків має бути зрозуміло, що ми зараз лише дряпаємо поверхню того, що генеративний ШІ може зробити для організацій і людей в них. Незабаром, наприклад, для таких систем може стати стандартною практикою створювати більшість або весь наш письмовий або графічний вміст — надавати перші чернетки електронних листів, листів, статей, комп'ютерних програм, звітів, публікацій у блогах, презентацій, відео, і так далі. Безсумнівно, розвиток таких можливостей матиме драматичні та непередбачені наслідки для права власності на контент і захисту інтелектуальної власності, але вони також, ймовірно, революціонізують знання та творчу роботу. Якщо припустити, що ці моделі штучного інтелекту продовжують розвиватися протягом

короткого періоду існування, ми навряд чи можемо уявити всі можливості та наслідки, які вони можуть створити.

3.7 Автоенкодера

AutoEncoders (AE) були представлені кілька років тому як потужний інструмент для стиснення зображень (або даних загалом). Ці архітектури складаються з кодера, який перетворює вхідні дані в прихований вектор меншої розмірності, як правило, за допомогою стандартних згорток і шарів об'єднання, і наступного декодера, який працює за допомогою таких операцій, як деконволюція та підвищення дискретизації, щоб спробувати реконструювати вхідне зображення. Ці дві CNN контролювалися за допомогою середньоквадратичної помилки (MSE), просто попиксельної різниці між вихідним і реконструйованим зображенням.

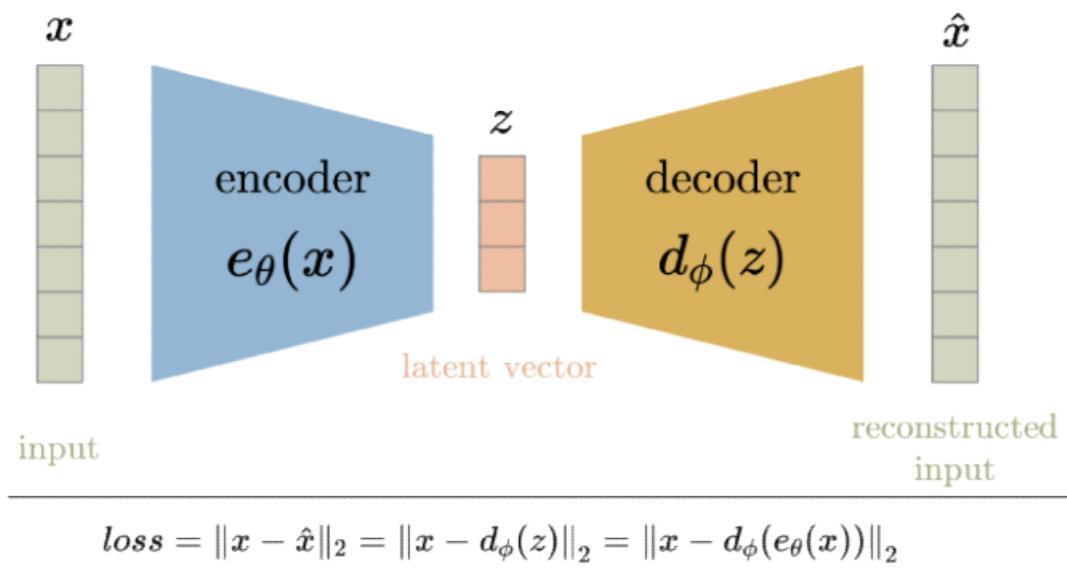


Рисунок 2.4 — Вихідним та реконструйоване зображення

Основна проблема AE полягає в тому, що латентний простір не регуляризований, що означає, що не кожен окремий латентний вектор дасть надійний результат. Це істотний недолік для синтезу зображення.

З цієї причини були представлені варіаційні автокодері (VAE). Основна відмінність полягає в тому, що кодер VAE повертає середнє значення та стандартне відхилення для кожного входу. Прихований вектор вибирається з цього розподілу та передається в декодер для реконструкції вхідних даних. Навчання тепер контролюється не лише втратою реконструкції, як у AE, але й втратою подібності, яка є розбіжністю KL між розподілом латентного простору та стандартним гаусовим (нульове середнє та одинична дисперсія). Більш чітко кажучи, мережа навчена не тільки реконструювати надійні зразки, але також створювати приховані вектори з нормального розподілу, переконавшись, що прихований простір рівномірно розподілений і не має значних проміжків між кластерами.

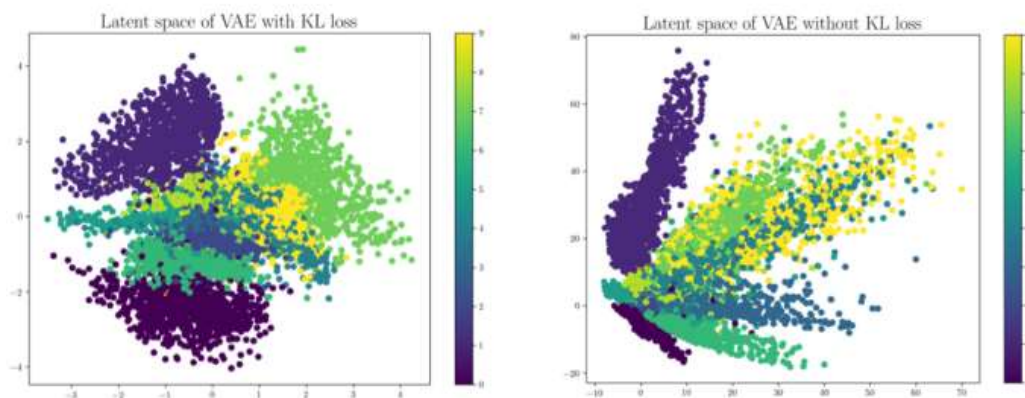


Рисунок 2.5 — Фундаментальна різниця між VAE та векторним квантовано-варіаційним автокодером

Фундаментальна відмінність між VAE та векторним квантовано-варіаційним автокодером (VQ-VAE) полягає в тому, що VAE вивчає безперервне приховане представлення, тоді як VQ-VAE вивчає дискретне приховане представлення. Ця ідея впливає з того факту, що більшість даних, з якими ми стикаємося в реальному світі, є дискретними. Наприклад, зображення містять окремі об'єкти з окремими характеристиками (колір, розмір, орієнтація, форма

тощо). Крім того, багато алгоритмів, наприклад Transformers, призначені для роботи з дискретними даними.

Основною новиною в архітектурі VQ-VAE є дискретна книга кодів, що вивчається, яка в основному є списком векторів із відповідними індексами. Цей компонент використовується для дискретизації вихідних даних кодера: кожен закодований вектор порівнюється з усіма векторами в кодовій книзі, і вибирається найближчий для евклідової відстані (формула argmin на зображенні нижче) для побудови квантованого представлення для подачі декодер.

Можна розглядати квантування як процес поділу векторів на групи, представлені кодовим словом, зазвичай отриманим за допомогою алгоритму кластеризації. Під час навчання вивчається словник кодових слів (кодова книга) та їхніх відповідників.

3.8 Векторні квантовані генеративні змагальні мережі (VQ-GAN)

Основна ідея полягає в тому, що для використання обчислювально дорогої операції самоуважності в синтезі з високою роздільною здатністю зображення мають бути певним чином виражені як послідовності. Замість використання пікселів або патчів як маркерів, кодер витягує кодування $Z_{\hat{}}$, яке потім квантується до Z_q за допомогою найближчого запису кодової книги (як у VQ-VAE). Потім декодер може реконструювати зображення, починаючи з квантування. Ця частина навчання, необхідна для вивчення подання кодової книги, майже ідентична до VQ-VAE, з двома основними відмінностями у використаних втратах. Дійсно, як уже було сказано в кінці останнього розділу, втрати, що використовуються у VQ-VAE, складаються з трьох членів, MSE плюс дві втрати на вирівнювання. Тут MSE замінюється втратою сприйняття, яка в основному є MSE, обчисленою не на двох зображеннях, а на деяких внутрішніх представленнях зображень. Наприклад, два зображення проходять через CNN, і характеристики n -го шару витягуються та порівнюються з MSE. Крім того, вони

додають змагальні втрати (типові втрати, що використовуються в GAN, де генератор і дискримінатор конкурують у мінімакській грі), щоб вирішити проблему розмиття VQ-VAE, з передбаченням реального/фальшивого не для всього зображення, а для окремого патчі.

Після цього етапу мережа вивчила кодову книгу, і тепер вона здатна виробляти Z_q із заданим зображенням, яке передається в декодер для реконструкції оригінального зображення.

Але для створення нових зразків необхідна друга фаза. Таким чином, мережа заморожена, і починається цей другий етап навчання, де ви подаєте зображення в кодер (який заморожений) і отримуєте квантування (Z_q), яке використовуватиметься як цільова послідовність для навчання трансформатора. По суті, Transformer (в даному випадку GPT-2) навчається виконанню завдання передбачення послідовності, використовуючи просту крос-ентропію для передбачення наступного токена. Трансформатор навчається на цільовому Z_q , тому, коли він спочатку бачить [1], правильний прогноз буде «42», потім він бачить [1, 42], а правильний прогноз буде «3» і так далі. На зображенні нижче ця концепція роз'яснена: за допомогою послідовності [1, 42, 3, 3, 94, 60, 22] трансформатор повинен передбачити «57».

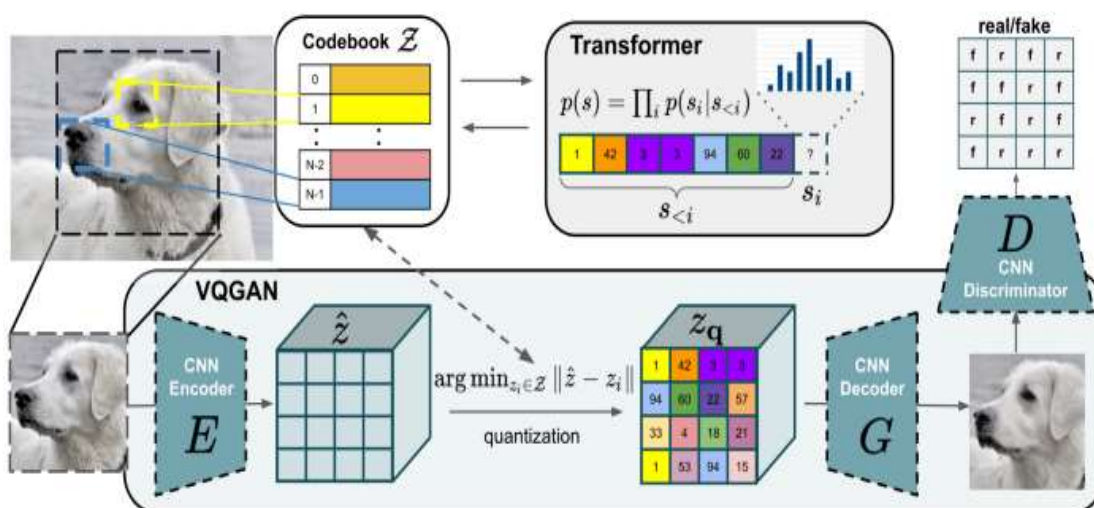


Рисунок 2.6 — Зображення архітектури VQ-VAE

3.9 Проведення моделювання за допомогою коду

Кроки для налаштування методу:

- Завантаження попередньо навченої моделі VQGANprint

('Downloading VQGAN model weights')

```
!curl -L 'https://heibox.uni-
heidelberg.de/d/8088892a516d4e3baf92/files/?p=%2Fckpts%2Flast.ckpt&dl=1' >
vqgan_im1024.ckpt
```

```
!curl -L 'https://heibox.uni-
heidelberg.de/d/8088892a516d4e3baf92/files/?p=%2Fconfigs%2Fmodel.yaml&dl=1' >
vqgan_im1024.yaml
```

#Установите требования к VQGAN

print('Installing requirements')

```
!git clone https://github.com/CompVis/taming-transformers &> /dev/null
```

```
!pip install einops omegaconf pytorch_lightning &> /dev/null
```

- 2. Налаштування VQGAN

```
import sys, einops, torch
```

```
sys.path.append('./taming-transformers')
```

```
from omegaconf import OmegaConf
```

```
from taming.models import cond_transformer, vqgan
```

```
from PIL import Image
```

```
from matplotlib import pyplot as plt
```

```
def load_vqgan_model(config_path, checkpoint_path):
```

```

config = OmegaConf.load(config_path)

if config.model.target == 'taming.models.vqgan.VQModel':
    model = vqgan.VQModel(**config.model.params)
    model.eval().requires_grad_(False)
    model.init_from_ckpt(checkpoint_path)

elif config.model.target == 'taming.models.cond_transformer.Net2NetTransformer':
    parent_model = cond_transformer.Net2NetTransformer(**config.model.params)
    parent_model.eval().requires_grad_(False)
    parent_model.init_from_ckpt(checkpoint_path)
    model = parent_model.first_stage_model

else:
    raise ValueError(f'unknown model type: {config.model.target}')

del model.loss

return model

print('Loading VQGAN model')

device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')

vqgan_model = load_vqgan_model('vqgan_im1024.yaml',
                              'vqgan_im1024.ckpt').to(device)

print('Dataset Prep')

!pip install openimages

!wget https://storage.googleapis.com/localized-
narratives/annotations/open_images_validation_captions.jsonl

import json

with open('/content/open_images_validation_captions.jsonl', 'r') as json_file:

```

```
    json_list = list(json_file)
for json_str in json_list[:5]:
    result = json.loads(json_str)

import urllib3
import boto3
import botocore
import concurrent.futures
import os
from tqdm.notebook import tqdm

def _download_single_image(arguments):
    if os.path.exists(arguments["dest_file_path"]):
        return
    try:
        with open(arguments["dest_file_path"], "wb") as dest_file:
            arguments["s3_client"].download_fileobj(
                "open-images-dataset",
                arguments["image_file_object_path"],
                dest_file,
            )
    except urllib3.exceptions.ProtocolError as error:
        _logger.warning(
            f"Unable to download image {arguments['image_file_object_path']} -- skipping",
            error,
        )
```

```
def download_images_by_id(image_ids,section, images_directory):
```

- Ми завантажимо зображення з AWS S3, тому нам знадобиться клієнт boto S3

```
s3_client = boto3.client(
```

```
's3',
```

```
config=botocore.config.Config(signature_version=botocore.UNSIGNED),
```

```
) device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
```

```
vqgan_model = load_vqgan_model('vqgan_im1024.yaml',
'vqgan_im1024.ckpt').to(device)
```

```
print('Dataset Prep')
```

```
!pip install openimages
```

```
!wget https://storage.googleapis.com/localized-
narratives/annotations/open_images_validation_captions.jsonl
```

```
import json
```

```
with open('/content/open_images_validation_captions.jsonl', 'r') as json_file:
```

```
    json_list = list(json_file)
```

- 4. Список аргументів функції, що повторюється
- 5. Який порівняємо з функцією завантаження

```
download_args_list = []
```

```
for image_id in image_ids:
```

```
    image_file_name = image_id + ".jpg"
```

```
    download_args = {
```

```
        "s3_client": s3_client,
```

```
        "image_file_object_path": section + "/" + image_file_name,
```

```

    "dest_file_path": os.path.join(images_directory, image_file_name),
}

download_args_list.append(download_args)

```

6. Використовую ThreadPoolExecutor для паралельного завантаження зображень
with concurrent.futures.ThreadPoolExecutor(max_workers=10) as executor:

Nikita, [09.09.2022 18:02]

- Використовую виконавця для зіставлення функції завантаження з набором аргументів, що ітеруються. list(tqdm(executor.map(_download_single_image, download_args_list),

```

total=len(download_args_list), desc="Downloading images"))

```

```
!mkdir -p ims
```

```
def get_openimages(n_images=2000):
```

```
    print('Downloading images')
```

```
    download_images_by_id([json.loads(s)['image_id'] for s in json_list[:n_images]],
'validation', 'ims/')

```

```
    data = [json.loads(s) for s in json_list[:n_images]]

```

```
    return data

```

Nikita, [09.09.2022 18:02]

```
type(vqgan_model)
```


ВИСНОВКИ

У цій роботі розглядається потенціал використання інформаційної технології обробки зображень, метою якої є пришвидшення роботи ШІ, виражених словами у тексті. Розроблений мною метод об'єднує дослідження з різних галузей, зокрема інформатики, інтелектуального аналізу даних, пошуку та вилучення тексту та комп'ютерної лінгвістики.

1. Проаналізовано існуючі підходи до побудови та використання штучного інтелекту.
2. Обрано тип архітектури нейронної мережі, на основі якого розроблено метод для покращення швидкості роботи.
3. У ході дослідження було розкрито основні класифікації моделей машинного навчання та їх особливості.
4. Було описано різні способи оцінки побудованої моделі.
5. Наприкінці було проаналізовано результати запропонованого рішення. У результаті було виявлено, що немає єдиного рішення для побудови моделі. Аналітики постійно спостерігають за різними сплесками в економічному середовищі, внаслідок чого побудовані раніше моделі стають неефективними. Крім того, великі компанії постійно працюють над створенням більш досконалих алгоритмів, використовуючи та впроваджуючи нові інструменти машинного навчання. Основною проблемою для бізнесу є вибір відповідної моделі, а також її валідацією та безперервним оновленням на свіжих даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мироненко С., Онищенко Є. Порівняльний аналіз методів для вирішення задачі сентимент аналізу тексту. *Computer-integrated technologies: education, science, production*. 2020. № 40. С. 140–145. URL: <https://doi.org/10.36910/6775-2524-0560-2020-40-21> (дата звернення: 25.12.2022).
2. Огляд методів обробки та аналізу текстів на природних мовах / С. І. Доценко та ін. *Інформаційно-керуючі системи на залізничному транспорті*. 2018. № 6. С. 26–32. URL: <https://doi.org/10.18664/ikszt.v0i6.151638> (дата звернення: 25.12.2022).
3. Пентилюк М. Наукові параметри аналізу тексту. *Дивослово*. 2017. № 9 (726), верес. С. 36–41.
4. Сентимент аналіз засобами нейронної мережі / К. Ялова та ін. *Математичне моделювання*. 2021. № 1(44). С. 30–37. URL: [https://doi.org/10.31319/2519-8106.1\(44\)2021.235906](https://doi.org/10.31319/2519-8106.1(44)2021.235906) (дата звернення: 27.12.2022).
5. Серажим К. Інформаційний аспект аналізу тексту. *Журналістика*. 2008. Вип. 7 (32). С. 40–48.
6. Шингалов Д. В., Тріщ О. В., Минайленко Р. М. Методи автоматичного аналізу настроїв в соціальних мережах : Thesis. 2017. URL: <http://dspace.kntu.kr.ua/jspui/handle/123456789/7489> (дата звернення: 27.12.2022).
7. A Study on text mining and text mining products. *International journal of science, technology and humanities*. 2014. Vol. 1, no. 1. P. 61–63. URL: <https://doi.org/10.26524/ijsth11> (date of access: 25.12.2022).
8. A survey on twitter sentiment analysis / E.-U. Rahman et al. *International journal of computer sciences and engineering*. 2018. Vol. 6, no. 11. P. 644–648. URL: <https://doi.org/10.26438/ijcse/v6i11.644648> (date of access: 27.12.2022).

9. Algorithmic amplification of politics on Twitter / F. Huszár et al. Proceedings of the national academy of sciences. 2021. Vol. 119, no. 1. URL: <https://doi.org/10.1073/pnas.2025334119> (date of access: 27.12.2022).
10. Altenberg B. A bibliography of publications relating to English computer corpora. English computer corpora. Berlin, Boston. URL: <https://doi.org/10.1515/9783110865967.355> (date of access: 25.12.2022).
11. Blake C. Text mining. Annual review of information science and technology. 2011. Vol. 45, no. 1. P. 121–155. URL: <https://doi.org/10.1002/aris.2011.1440450110> (date of access: 25.12.2022).
12. Drus Z., Khalid H. Sentiment analysis in social media and its application: systematic literature review. Procedia computer science. 2019. Vol. 161. P. 707–714. URL: <https://doi.org/10.1016/j.procs.2019.11.174> (date of access: 27.12.2022).
13. Duncombe C. The politics of twitter: emotions and the power of social media. International political sociology. 2019. Vol. 13, no. 4. P. 409–429. URL: <https://doi.org/10.1093/ips/olz013> (date of access: 27.12.2022).
14. Eriksson M. Lessons for crisis communication on social media: a systematic review of what research tells the practice. International journal of strategic communication. 2018. Vol. 12, no. 5. P. 526–551. URL: <https://doi.org/10.1080/1553118x.2018.1510405> (date of access: 27.12.2022).
15. Finegan E., Johansson S. Computer corpora in english language research. Language. 1984. Vol. 60, no. 1. P. 190. URL: <https://doi.org/10.2307/414219> (date of access: 25.12.2022).
16. Garcia M. B. Sentiment analysis of tweets on coronavirus disease 2019 (COVID-19) pandemic from metro manila, philippines. Cybernetics and information technologies. 2020. Vol. 20, no. 4. P. 141–155. URL: <https://doi.org/10.2478/cait-2020-0052> (date of access: 27.12.2022).
17. Google trends. URL: <https://trends.google.com/trends/> (date of access: 25.12.2022).

18. Karpicke J. D., Grimaldi P. J. Retrieval-Based learning: a perspective for enhancing meaningful learning. *Educational psychology review*. 2012. Vol. 24, no. 3. P. 401–418. URL: <https://doi.org/10.1007/s10648-012-9202-2> (date of access: 25.12.2022).
19. Korney A. O., Kryuchkova E. N. Semantic-statistical algorithm for determining the categories of aspects in the problems of sentiment analysis. *IZVESTIYA SFedU. ENGINEERING SCIENCES*. 2021. No. 6. P. 66–74. URL: <https://doi.org/10.18522/2311-3103-2020-6-66-74> (date of access: 27.12.2022).
20. Kumar A., Sharma S., Singh D. Sentiment analysis on twitter data using a hybrid approach. *International journal of computer sciences and engineering*. 2019. Vol. 7, no. 5. P. 906–911. URL: <https://doi.org/10.26438/ijcse/v7i5.906911> (date of access: 27.12.2022).

Додаток Б

ДЕМОНСТРАЦІЙНІ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



Кафедра інженерії програмного забезпечення

МАГІСТЕРСЬКА РОБОТА

**«РОЗРОБКА МЕТОДУ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ
ТЕКСТОВОГО ОПИСУ З ВИКОРИСТАННЯМ ШТУЧНОГО
ІНТЕЛЕКТУ»**

Виконав: студент групи ПДМ-62 Сербулов Нікіта Євгенійович

Керівник: к.т.н., доц., доц. кафедри ІІЗ, Трінтіна Наталя Альбертівна

Київ - 2022

МАТЕРІАЛИ

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

2

Мета роботи: пришвидшення процесу отримання та аналізу даних штучного інтелекту для безперервного навчання з метою підвищення ефективності побудованих моделей.

Об'єкт дослідження: моделювання поведінки штучного інтелекту.

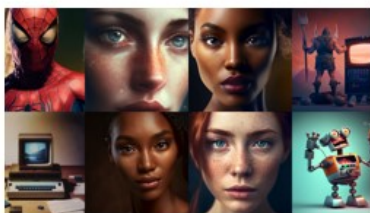
Предмет дослідження: дослідження моделі штучного інтелекту для пришвидшення роботи його алгоритмів

АНАЛІЗ ІСНУЮЧИХ ІТ-РІШЕНЬ ТА ЇХ МОДЕЛЕЙ

3



DALL-E - це моделі глибокого навчання, розроблені OpenAI для створення цифрових зображень із описів природною мовою.



DeepDream - це програма комп'ютерного бачення, створена інженером Google Олександром Морвінцевим, яка використовує згорточну нейронну мережу для пошуку та покращення шаблонів у зображеннях за допомогою алгоритмічної парейдолії.

Midjourney — це незалежна дослідницька лабораторія, яка розробляє власну програму штучного інтелекту, яка створює зображення з текстових описів.

МОДЕЛЬ З ПЕРЕТВОРЕННЯ ТЕКСТУ НА ЗОБРАЖЕННЯ

4

Штучний інтелект — здатність інженерної системи (англ. engineered system) обробляти, застосовувати та вдосконалювати здобуті знання та вміння

Штучні нейронні мережі — це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Такі системи навчаються задач (поступально покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу. Наприклад, у розпізнаванні зображень вони можуть навчатися ідентифікувати зображення, які містять котів, аналізуючи приклади зображень, мічені як «кіт» і «не кіт», і використовуючи результати для ідентифікування котів в інших зображеннях. Вони роблять це без жодного апіорного знання про котів, наприклад, що вони мають хутро, хвости, вуса та котоподібні писки. Натомість, вони розвивають свій власний набір доречних характеристик з навчального матеріалу, який вони оброблюють.



НЕЙРОННА МЕРЕЖА GPT

7

GPT:

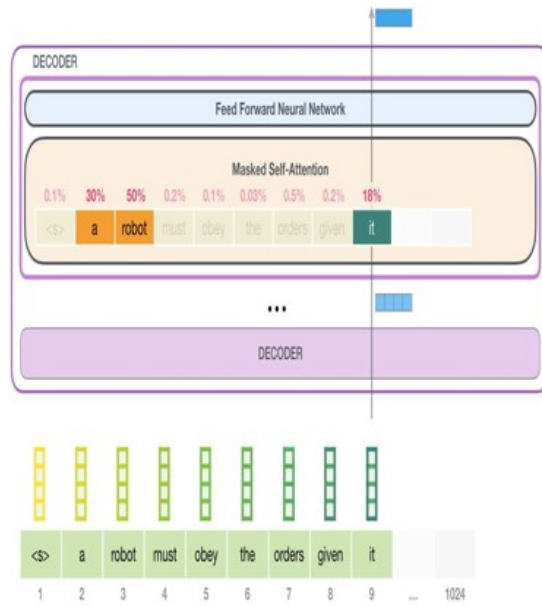
GPT намагається передбачити послідовність токенів на основі даної їй послідовності. Модель є архітектурою Transformers, що складається тільки з Декодера. GPT навчається фокусувати увагу на попередніх словах, які є найбільш важливими для передбачення наступного слова в реченні, використовуючи механізм уваги.

Attention:

Механізм Attention є аналогом когнітивної уваги та дозволяє знаходити зв'язки між токенами, а також передбачати продовження цих токенів.

Для кожного токена створюється набір векторів:

1. Key - значимість токена в послідовності, якщо ми дивимось ззовні.
2. Query - значимість токена в послідовності, якщо ми бачимо з цього токена.
3. Value - репрезентація токена

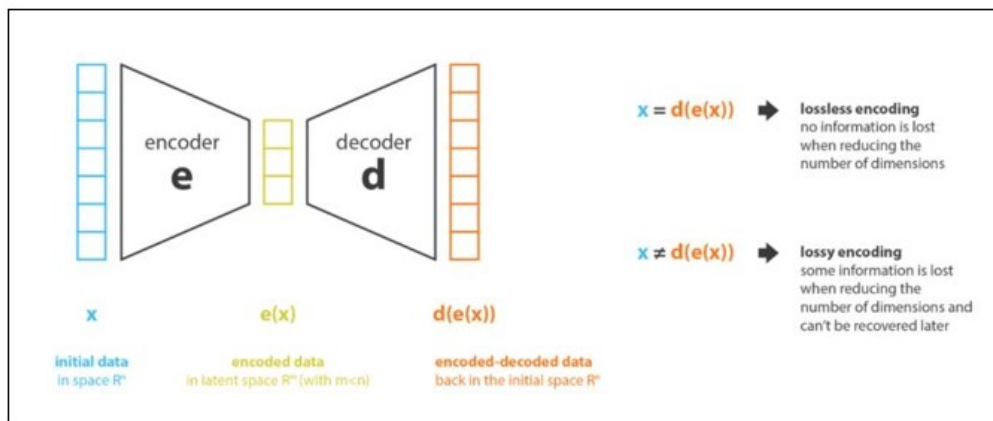


АВТОЕНКОДЕР АРХІТЕКТУРИ VAE

6

Автоенкодер:

На вході моделі – випадковий шум із бажаним значенням розміру вихідного сигналу. Спочатку зразок редукується в латентний простір меншого розміру. Для цього автори використовували архітектуру VAE, що складається з двох частин - кодера та декодера. Кодер використовується під час навчання для конвертації зразка у зменшене латентне подання та передає його як вхід до наступного блоку. При виведенні, що пройшли денойзинг згенеровані зразки піддаються зворотній дифузії і трансформуються назад у свої початкові розміри латентного простору.



ДВУХ ЕТАПНІ ДИФУЗНІ МОДЕЛІ

5

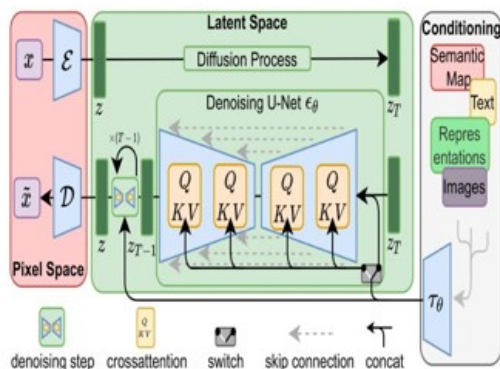
Дифузні моделі складаються з двох етапів:

- **Пряма дифузія (Forward Diffusion)** — мапування даних із шумом шляхом поступового внесення перешкод у вихідну інформацію. Формально це досягається простим стохастичним (випадковим) процесом, який починається з вихідного зразка та ітеративно генерує зашумлені екземпляри, використовуючи просте гаусове ядро дифузії. Цей процес використовується лише під час навчання, і під час виведення не застосовується.
- **Параметризований реверс** – відкочує назад пряму дифузію та здійснює процес ітеративного денойзингу. Він є синтезом даних і навчений генерувати інформацію шляхом перетворення випадкового шуму в реалістичні дані.

Прямий та зворотний (реверсивний) процеси вимагають послідовного повторення тисяч кроків, внесення та зменшення шуму, що робить весь процес повільним та важким для обчислювальних ресурсів.

Щоб забезпечити можливість навчання на обмежених ресурсах за збереження якості та гнучкості, творці Stable Diffusion застосували метод, описаний у статті. Замість використання фактичного піксельного простору вони застосували процес дифузії над латентною областю нижчої розмірності.

Наприклад, автокодер, що використовується в Stable Diffusion, має коефіцієнт редукції (зменшення) 8. Це означає, що зображення форми (3, 512, 512) стає (3, 64, 64) у латентному просторі, що вимагає $8 \times 8 = 64$ рази менше за пам'ять.



VQ-GAN (Vector Quantized Generative Adversarial Networks)

8

VQ-GAN здатна стискати зображення у сітку векторів (tokenів) та реконструювати назад у зображення. Нейросітка складається з енкодера, декодера та дискримінатора.

VQ-GAN є поєднанням ідей VQVAE та GAN.

VQ-VAE (Vector Quantized Variational Autoencoder)

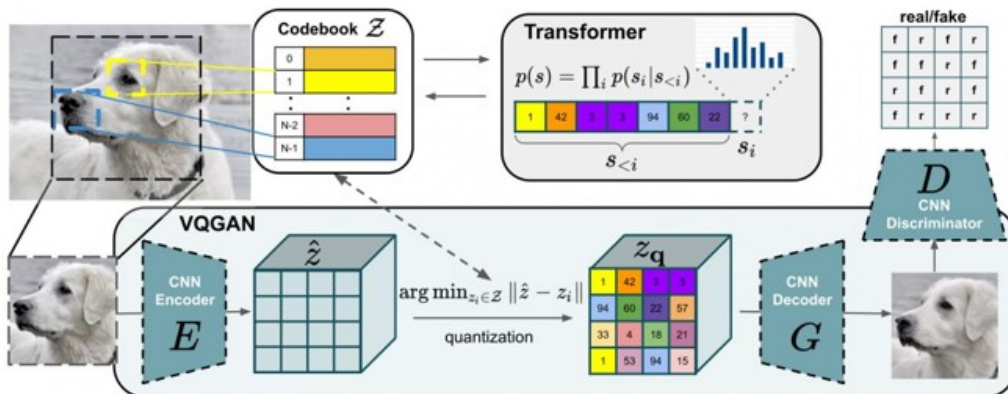
Ідея VQ-VAE в тому, щоб навчитися ефективно стискати зображення в більш низькорозмірний прихований простір і розтискати в прихований простір зображення з найменшими втратами.

Енкодер бере зображення розміром 512×512 і стискає (найчастіше звичайними згортковими мережами) до 256×1 , шукає найближчий вектор від прихованого простору, а потім декодер намагається такий маленький вектор розтиснути назад у зображення 512×512 .

VQ-VAE + GAN = VQ-GAN

Енкодер так само приймає на вхід зображення і кодує його в стислий вектор, після чого піднімає на найближчого сусіда в прихованому просторі, а Декодер розтискає цей найближчий вектор зображення.

Тут у гру входить частинка GAN, зокрема дискримінатор. Дискримінатор у процесі навчання порівнює, наскільки вихідне зображення схоже на реконструйоване та повертає градієнт втрат реконструкції, це допомагає надалі зрозуміти енкодеру та декодеру, як обдурити Дискримінатор та в рази підвищити якість кінцевих зображень.



РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

9

1. Енкодер перетворює зображення X на сітку прихованих векторів

$$x' = G(z_q) = G(\mathbf{q}(E(x)))$$

2. Кожен вектор у сітці піддається квантуванню з використанням функції $q(\hat{z})$

$$z_q = \mathbf{q}(\hat{z}) := \left(\arg \min_{z_i \in Z} \|\hat{z}_{uv} - z_i\| \right) \in \mathbb{R}^{h \times w \times d}$$

3. Декодер реконструює квантовану сітку

$$\hat{x} = G(z_q) = G(\mathbf{q}(E(x)))$$

4. Ця функція втраг вираховує, як добре моделі вдалося реконструювати оригінал

$$\mathcal{L}_{VQ}(E, G, Z) = \|x - \hat{x}\|^2 + \|sg[E(x)] - z_q\|_2^2 + \|sg[z_q] - E(\hat{x})\|_2^2$$

5. Дискримінатор дозволяє досягти кращої реконструкції зображень, навіть із дуже стиснутої сітки векторів.

$$\mathcal{L}_{GAN}(\{E, G, Z\}, D) = [\log D(x) + \log(1 - D(\hat{x}))]$$

6. Тепер за допомогою цієї функції ми можемо навчити наш декодер:

$$Q^* = \arg \min_{E, G, Z} \max_D \mathbb{E}_{x \sim p(x)} [\mathcal{L}_{VQ}(E, G, Z) + \lambda \mathcal{L}_{GAN}(N, D)]$$

ВИСНОВКИ

10

1. Проаналізовано існуючі підходи до побудови та використання штучного інтелекту.
2. Обрано тип архітектури нейронної мережі, на основі якого розроблено метод для покращення швидкості роботи.
3. У ході дослідження було розкрито основні класифікації моделей машинного навчання та їх особливості.
4. Було описано різні способи оцінки побудованої моделі.
5. Наприкінці було проаналізовано результати запропонованого рішення. У результаті було виявлено, що немає єдиного рішення для побудови моделі. Аналітики постійно спостерігають за різними сплесками в економічному середовищі, внаслідок чого побудовані раніше моделі стають неефективними. Крім того, великі компанії постійно працюють над створенням більш досконалих алгоритмів, використовуючи та впроваджуючи нові інструменти машинного навчання. Основною проблемою для бізнесу є вибір відповідної моделі, а також її валідацією та безперервним оновленням на свіжих даних.

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

11

Тези доповідей на конференціях:

1. Сербулов Н.Е. Трансформвання робочого процесу за допомогою штучного інтелекту// XV Конференції «Сучасні інфокомунікаційні технології», 9 грудня 2022 року, Державний університет телекомунікації, Київ, Україна.
2. Сербулов Н.Е. Використання нейромереж у процесі інтелектуального аналізу даних/Міжнародна науково-практична конференція «Актуальні питання забезпечення кібербезпеки та захисту інформації», 22-23 січня 2023, Національний Авіаційний Університет Київ, Україна.

ДЯКУЮ ЗА УВАГУ!