

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту інформації
(повна назва)

Кафедра медіаінженерії та інформаційних радіоелектронних систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Створення фотореалістичного контенту за допомогою нейромереж.
(тема)

Виконав:
студент 2 курсу, групи СТМм-22-2
Сірокуров В.І.
(прізвище, ініціали)

Спеціальність 171 Електроніка
(код і повна назва спеціальності)

Тип програми освітньо-професійна
Освітня програма Системи, технології і комп'ютерні засоби мультимедіа
(повна назва освітньої програми)

Керівник доц. Іванова О.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри Володимир КАРТАШОВ
(підпис)

2023 р.

Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту інформації

Кафедра медіаінженерії та інформаційних радіоелектронних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 171 Електроніка

(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Системи, технології і комп'ютерні засоби мультимедіа

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Сірокурову Валерію Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Створення фотореалістичного контенту за допомогою нейромереж.

затверджена наказом по університету від " 20 " 11 _____ 2023 р. № 1371 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 10.01.2024 р.

3. Вихідні дані до роботи Провести аналіз основ роботи нейромереж. Розглянути види нейромереж та процес їх навчання. Провести теоретичний аналіз задачі генерації зображень. Розглянути принципи організації генеративно-змагальних мереж та результатів їхньої роботи. Обрати платформу для досліджень. Дослідити принципи формування запиту до нейромережі, а саме: вибір об'єкту, уособлення об'єкту, кількість об'єктів, конкретність запитів, генерація абстракцій, проблема генерації кінцівок, додавання деталей, налаштування освітлення, налаштування параметрів камери. Провести дослідження роботи з фотопортретами. Зробити висновки.

4. Перелік питань, що потрібно опрацювати в роботі _____

Вступ

1 Аналіз передумов і механізмів для введення просторової обробки звукових сигналів.

2 Теоретичний аналіз методів просторової обробки в зведенні і майстерингу звуку.

3 Практичні прийоми просторової обробки при зведенні і майстерингу.

Висновки

Перелік посилань

Додатки

5. Перелік графічного матеріалу із зазначенням обов'язкових креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Постановка задачі (1 слайд).
2. Основи роботи нейромереж (1 слайд).
3. Види нейромереж (1 слайд).
4. Навчання нейромереж (1 слайд).
5. Задача генерації зображень (1 слайд).
6. Спрощене описання генерації (1 слайд).
7. Генеративно-змагальна мережа (1 слайд).
8. Результати генеративно-змагальних мереж (1 слайд).
9. Платформа для досліджень (1 слайд).
10. Вибір об'єкту (1 слайд).
11. Уособлення об'єкту (1 слайд).
12. Кількість об'єктів (1 слайд).
13. Конкретність запитів (1 слайд).
14. Генерація абстракцій (1 слайд).
15. Проблема генерації кінцівок (1 слайд).
16. Додавання деталей (1 слайд).
17. Налаштування освітлення (1 слайд).
18. Налаштування параметрів камери (1 слайд).
19. Робота з фотопортретами (1 слайд).
20. Зміна віку і стилю (1 слайд).
21. Висновки (1 слайд).

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|--------------------------------|----------|
| 1 | Аналітичний огляд літератури | 25.09.23–01.10.23 | |
| 2 | Теоретичний аналіз нейромереж | 02.10.23–11.10.23 | |
| 3 | Навчання роботі з нейромережею | 12.10.23–10.11.23 | |
| 4 | Експериментальне дослідження нейромережі | 11.11.23–03.12.23 | |
| 5 | Обробка результатів дослідження | 04.12.23–17.12.23 | |
| 6 | Графічна частина роботи | 18.12.23–17.12.23 | |
| 7 | Перевірка керівником | 18.12.23–30.12.23 | |
| 8 | Перевірка на академічний плагіат | 02.01.24–05.01.24 | |
| 9 | Перевірка завідувачем кафедри, рецензування | 06.01.24–09.01.24 | |
| | | | |
| | | | |

Дата видачі завдання _____ 25.09.2023 р. _____

Студент _____ Валерій СИРОКУРОВ _____

Керівник роботи _____ Олена ІВАНОВА _____

(підпис)

(підпис)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 68 сторінок, 42 рисунки, 30 джерел.

ГЕНЕРАЦІЯ ЗОБРАЖЕННЯ, ГЕНЕРАТИВНО-ЗМАГАЛЬНІ МЕРЕЖІ, НЕЙРОМЕРЕЖА, ЗГОРТКОВА МЕРЕЖА, НЕЙРОМЕРЕЖА ПРЯМОГО ПОШИРЕННЯ, ПЕРЦЕПТРОН

Метою кваліфікаційної роботи є теоретичний аналіз основ побудови, функціонування, навчання та оцінки якості навчання нейронних мереж для обробки і генерації фотореалістичних зображень, а також практичне дослідження особливостей роботи і результатів, що дають на сьогоднішній день такі нейромережі.

В роботі проведено аналіз основ роботи нейромереж. Розглянуто види нейромереж та процес їх навчання. Проведено теоретичний аналіз задачі генерації зображень. Розглянуто принципи організації генеративно-змагальних мереж та результатів їхньої роботи. Обрано платформу для досліджень. Досліджено принципи формування запиту до нейромережі, а саме: вибір об'єкту, уособлення об'єкту, кількість об'єктів, конкретність запитів, генерація абстракцій, проблема генерації кінцівок, додавання деталей, налаштування освітлення, налаштування параметрів камери. Проведено дослідження роботи з фотопортретами. Зроблено висновки.

ABSTRACT

Explanatory note to the qualification work: 68 pages, 42 drawings, 30 sources.

IMAGE GENERATION, GENERATIVE-COMPETITIVE NETWORKS, NEURAL NETWORK, CONVOLUTIONAL NETWORK, DIRECT PROPAGATION NEURAL NETWORK, PERCEPTRON

The purpose of the qualification work is a theoretical analysis of the foundations of construction, operation, training and evaluation of the quality of training of neural networks for processing and generating photorealistic images, as well as a practical study of the peculiarities of the work and results that such neural networks provide today.

The paper analyzes the basics of neural networks. The types of neural networks and the process of their training are considered. A theoretical analysis of the problem of image generation was carried out. The principles of the organization of generative-competitive networks and the results of their work are considered. A research platform is selected. The principles of forming a request to a neural network were studied, namely: object selection, object impersonation, number of objects, specificity of requests, generation of abstractions, problem of limb generation, addition of details, lighting settings, camera settings. A study of working with photo portraits was conducted. Conclusions have been made.

ЗМІСТ

| | |
|--|-----------|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів..... | 8 |
| Вступ..... | 9 |
| 1 АНАЛІЗ МЕТОДІВ ПОБУДОВИ, ПРИНЦИПІВ РОБОТИ, НАВЧАННЯ ТА ОСОБЛИВОСТЕЙ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ..... | 11 |
| 1.1 Визначення нейромереж і історія їх появи..... | 11 |
| 1.2 Принцип роботи нейронної мережі..... | 14 |
| 1.3 Структури і види нейронних мереж..... | 16 |
| 1.4 Навчання нейронних мереж..... | 19 |
| 1.5 Завдання і сфери застосування нейромереж..... | 20 |
| 1.6 Критерії якості функціонування нейронних мереж..... | 22 |
| 1.7 Висновки по розділу 1..... | 26 |
| 2 ПРИНЦИПИ РОБОТИ НЕЙРОМЕРЕЖ ДЛЯ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ..... | 28 |
| 2.1 Основи генерації зображень..... | 28 |
| 2.1.1 Дискримінатори зображення..... | 28 |
| 2.1.2 Генератори зображення..... | 29 |
| 2.2 Тренування генератора..... | 31 |
| 2.3 Генеративні змагальні мережі..... | 32 |
| 2.4 Алгоритм роботи генеративно-змагальної мережі GAN..... | 34 |
| 2.5 Проблеми тренування..... | 35 |
| 2.6 Image-to-image трансляція..... | 37 |
| 2.7 Приклад розробки image-to-image транслятора..... | 38 |
| 2.8 Навчання pix2pixHD..... | 40 |
| 2.9 Нейромережа SPADE..... | 42 |
| 2.10 Висновки по розділу 2..... | 44 |

| | |
|---|----|
| 3 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ГЕНЕРАЦІЇ ФОТОРЕАЛІСТИЧНИХ ЗОБРАЖЕНЬ НЕЙРОМЕРЕЖАМИ..... | 46 |
| 3.1 Загальні правила складання запитів..... | 46 |
| 3.2 Вибір об'єкта..... | 46 |
| 3.3 Додавання деталей..... | 51 |
| 3.4 Визначення стилю..... | 53 |
| 3.5 Огляд найпопулярніших команд для роботи з Openjourney..... | 55 |
| 3.6 Експеримент з фотореалістичними портретами..... | 57 |
| 3.7 Висновки по розділу 3..... | 59 |
| Висновки..... | 60 |
| Перелік посилань..... | 65 |
| ДОДАТКИ..... | 69 |
| Додаток А. Графічний матеріал..... | 70 |
| Додаток Б. Відомість кваліфікаційної роботи..... | 91 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

ЗНМ – згорткова нейронна мережа;

НМ – нейронна мережа;

Синапси – сполучення, які використовуються для того, щоб надсилати повідомлення між нейронами;

ШНМ – штучна нейронна мережа;

ПФ – передавальна функція;

ANN – artificial neural network – штучна нейронна мережа;

BPNN – back propagation neural network – нейронна мережа зі зворотним поширенням

CNN – convolutional neural network – згорткова нейронна мережа;

CV – Computer Vision – комп'ютерне бачення;

DL – deep learning – глибоке навчання;

DNN – deep neural network – глибока нейромережа;

FNN – Fedforward neural network – нейромережа прямого поширення;

RNN – recurrent neural network – рекурентна нейромережа;

NN – Neural Network – нейромережа;

NLP – natural language processing – обробка натурального мовлення;

ML – machine learning – машинне навчання;

MLP – multilayer perceptron – багатошаровий перцептрон;

net – сигнал суматора нейрону;

threshold – порог функції активації.

ВСТУП

Перші нейронні мережі у вигляді комп'ютерного коду були реалізовані ще в 1950-х роках. Тоді вони вирішували прості задачі класифікації, на більш складні завдання не вистачало обчислювальної потужності. Новий поштовх нейромережі отримали в 2010-х роках, коли почали вирішувати задачі аналізу великих об'ємів даних і прогнозування, розпізнавання та синтезу тексту або мови, комп'ютерного зору і багатьох інших задач.

Не оминули нейромережі і мультимедіа технології. Сьогодні багато програмних продуктів включають спеціальні модулі зі штучним інтелектом, що дозволяють виконувати не тільки обробку звукових, графічних та відеоданих за допомогою нейромереж, але й суттєво модифікувати мультимедійний контент, прибираючи або добавляючи окремі об'єкти на зображенні, або сильно видозмінюючи головний об'єкт на фото. Причому результати такої обробки при правильному підході виглядають дуже реалістичними. Вершиною складності є генерація нейромережею власного графічного контенту на основі запиту.

Генеративні нейромережі працюють за принципом «вигадали запит, ввели його та отримали відповідь». Але все не так просто: якщо прописати абстрактне питання або перевантажити його деталями, нейромережа заплутається і у відповідь видасть «кашу». Сучасний спеціаліст в сфері мультимедіа має знати і вміти, як правильно прописувати запити та отримувати коректні відповіді.

Отже, штучний інтелект можна розглядати як повноцінний інструмент створення мультимедійного контенту, поряд з іншими, класичними методами обробки. Як показує практика, сьогодні не існує точних рекомендацій для побудови структури, процесу навчання і критеріїв якості нейронних мереж. Робота з нейромережею лишається дуже суб'єктивним і творчим процесом. Тут допомагає досвід та базові знання про функціонування даного інструменту.

Метою кваліфікаційної роботи є теоретичний аналіз основ побудови, функціонування, навчання та оцінки якості навчання нейронних мереж для обробки і генерації фотореалістичних зображень, а також практичне дослідження особливостей роботи і результатів, що дають на сьогоднішній день такі нейромережі.

1 АНАЛІЗ МЕТОДІВ ПОБУДОВИ, ПРИНЦИПІВ РОБОТИ, НАВЧАННЯ ТА ОСОБЛИВОСТЕЙ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ

1.1 Визначення нейромереж і історія їх появи

Нейронна мережа (neural network) – це комп'ютерний алгоритм, здатний обробляти великі обсяги даних, імітуючи діяльність людського мозку. Як і людина, нейромережа вивчає нові предмети, робить висновки та надалі використовує отриману інформацію [1]. Нейромережі є математичними моделями, створеними з урахуванням структури і принципів функціонування біологічних нейронних мереж, по аналогії з людським мозком.

Нервову систему людини утворюють нейрони – клітини, які отримують інформацію та транслюють її у вигляді імпульсів (рис.1.1) [2]. Основна частина нейрона – аксон, а довгий відросток на його кінці зветься дендрит, він виконує роль своєрідного дроту при передачі інформації від одного нейрона до іншого. Таким чином мозок, транслюючи інформацію, керує всіма діями людини.

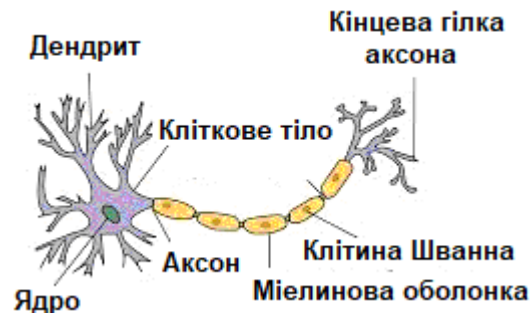


Рисунок 1.1 – Типова структура нейрона

На основі відповідного принципу працюють і комп'ютерні нейронні мережі, що стали цифровою моделлю людського мозку. Головна їх особливість – здатність до навчання. Стандартні комп'ютерні програми припускають, що алгоритм їм пише людина, тобто задає певний набір дій, які мають виконати комп'ютери. При використанні нейромережі не потрібно казати їй, як вирішити завдання. Достатньо задати вступні дані, а способам

вирішення задач нейронна мережа на основі штучного інтелекту навчається сама, виявляючи закономірності та виявляючи на їх основі способи вирішення задач.

Для побудови штучної нейронної мережі використовуватимемо ту саму структуру. Як і біологічна нейронна мережа, штучна складається з нейронів, що взаємодіють між собою, проте є спрощеною моделлю. Так, наприклад, штучний нейрон, з яких складається ІНС, має набагато простішу структуру: він має кілька входів, на яких він приймає різні сигнали, перетворює їх і передає іншим нейронам. Іншими словами, штучний нейрон (рис.1.2) – це така функція $\vec{R} \rightarrow R$, яка перетворює кілька вхідних параметрів в один вихідний [2].

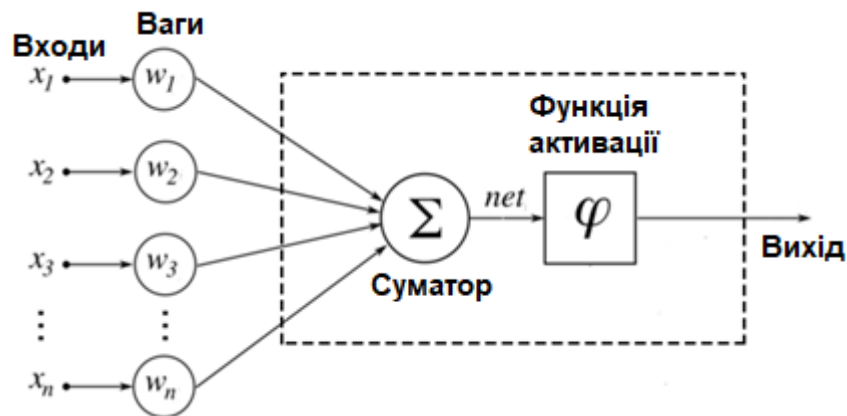


Рисунок 1.2 – Схема штучного нейрона

Як видно на рис.1.2, нейрон має n входів x_i , у кожного з якого є вага w_i , на який множиться сигнал, що проходить по зв'язку. Після цього зважені сигнали $x_i \cdot w_i$ направляються до суматора, який поєднує всі сигнали у зважену суму. Цю суму також називають net . Таким чином, [3]

$$net = \sum_{i=1}^n w_i x_i = w^T x. \quad (1.1)$$

Просто так передавати виважену суму net на вихід немає сенсу – нейрон повинен її якось обробити та сформувавати адекватний вихідний сигнал.

Для цих цілей використовують функцію активації, яка перетворює зважену суму в якесь число, яке буде виходом нейрона. Функція активації позначається $\varphi(net)$. Таким чином, виходів штучного нейрона є $\varphi(net)$.

Для різних типів нейронів використовують різні функції активації, але одними з найпопулярніших є [3]:

– функція одиничного стрибка (рис.1.3, а) [3]:

$$\begin{aligned} \text{якщо } net > \text{threshold}, \varphi(net) &= 1, \\ \text{інакше, } \varphi(net) &= 0; \end{aligned} \quad (1.2)$$

– сигмоїдальна функція (рис.1.3, б) [3]:

$$\varphi(net) = \frac{1}{1 + \exp(-a \cdot net)}, \quad (1.3)$$

де параметр a характеризує ступінь крутості функції;

– гіперболічний тангенс (рис.1.3, в) [3]:

$$\varphi(net) = \tanh(a \cdot net), \quad (1.4)$$

де параметр a також визначає ступінь крутості графіка функції;

– Rectified linear units (ReLU, рис.1.3, г) [3]:

$$\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} = \max(x, 0). \quad (1.5)$$

Спроби математично описати мережу нейронів робилися ще 1940-ті роки. Ідею створення нейронних мереж вперше запропонували дослідники з університету Чикаго Уоррен Маккалоу і Уолтер Пітс.

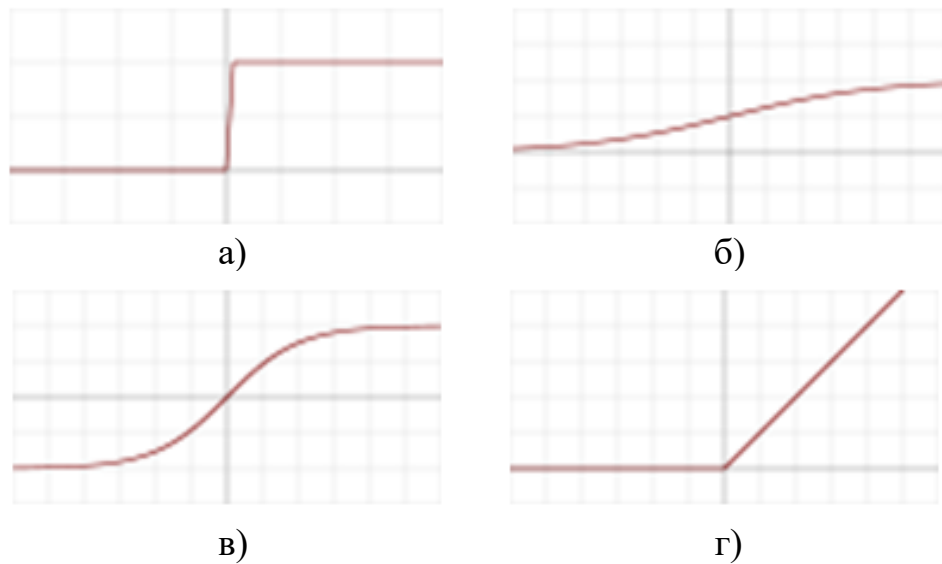


Рисунок 1.3 – Найпопулярніші функції активації $\varphi(net)$

У 1950-ті роки ця математична модель була відтворена психологом університету Корнелл Френком Розенблаттом за допомогою комп'ютерного коду. Розенблатт був автором перцептронів – прототипу сучасних неймереж. Навіть така елементарна структура у роки могла навчатися і самостійно вирішувати прості завдання [4].

Однак для створення моделей потужних мереж на той момент було недостатньо, тому їхній розвиток сповільнився. Воно відновилося лише у 2010-ті роки, з розвитком комп'ютерних технологій та появою потужних комп'ютерів. Наступним етапом розвитку стала поява неймереж зі штучним інтелектом [4].

1.2 Принцип роботи нейронної мережі

Головна відмінність неймережеских моделей від класичних полягає у їх структурі. Основні елементи, з яких він складається – штучні нейрони та зв'язки між ними (рис.1.4) [5]. Штучні нейрони також називаються словом «вузли» – елементарні обчислювальні одиниці, пов'язані між собою. Вони є нелінійними функціями з одним аргументом. Нейрон отримує загальну

інформацію, здійснює обчислення та передає дані далі. Кожен нейрон має два параметри: вхідні дані (input data) та вихідні дані (output data).



Рисунок 1.4 – Складові частини нейронних мереж

Синапси – сполучення, які використовуються для того, щоб надсилати повідомлення між нейронами. Кожне з них має певну вагу. Це число, яке множиться значення вхідного сигналу, коефіцієнт, що визначає взаємозв'язок між нейронами. Чим це значення вище, тим важливішим є зв'язок між вузлами.

Якщо значення ваги на вихід перевищено, вузол активується та надсилає дані наступному нейрону. Якщо показники значень нижче, передача даних немає – у цьому разі говорять про випереджувальні зв'язки, коли дані проходять лише одному напрямі. Таким чином, проходячи через синапси, сигнал слабшає, посилюється або залишається рівним і постійним, що зрештою впливає на результат.

Отже мозок системи – матриця вагових коефіцієнтів, тобто всі ваги нейронної мережі. Саме завдяки їм інформація обробляється та передається далі.

Шари. Нейронів у нейромережі багато, тому вони об'єднуються у шари (рис.1.5) [5]:

- вхідний, куди надходять дані. Вони можуть мати будь-який формат – файли, тексти, музика, картинки, відео та інші;
- приховані, в яких виробляються обчислення та обробка. Зазвичай прихованих шарів не більше трьох;
- вихідний – звідси виходять результати.



Рисунок 1.5 – Шарова структура нейронної мережі

Таким чином, чим більше шарів у нейронній мережі, тим складніше завдання, з якими вона може справлятися.

Принцип роботи нейронної мережі полягає в наступному. Інформація у вигляді тексту, зображень чи іншому форматі надходить на зовнішній шар. Нейрони зовнішнього шару розпізнають її, класифікують та передають далі.

У прихованому шарі відбувається основна робота. Прихованих шарів може бути кілька, іноді їхня кількість доходить до мільйона. При проходженні через приховані шари попередні значення даних множаться на вагу зв'язку, після чого підсумовуються результати.

Відповідь мережі формується у вихідному шарі. Формат відповіді може бути будь-яким.

Якщо мережа не навчена, класифікація вагів відбувається випадково. Значимість кожного нейрона підвищується у процесі навчання, якщо вони призводять до правильного рішення. Цей складний алгоритм можна порівняти з роботою людського мозку: він навчається чогось нового, завдяки чому зміцнюються нейронні зв'язки [6].

1.3 Структури і види нейронних мереж

Як правило, у більшості нейронних мереж є так званий вхідний шар, який виконує лише одне завдання – розподіл вхідних сигналів по іншим

нейронам. Нейрони цього шару не виробляють жодних обчислень. В іншому нейронній мережі поділяються на основні категорії, наведені нижче.

Одношарова нейронна мережа (Single-layer neural network) – мережа, в якій сигнали від вхідного шару відразу подаються на вихідний шар, який і перетворює сигнал і відразу видає відповідь.

Як видно зі схеми одношарової нейронної мережі, представлені на рис.1.6 [7], сигнали x_1, x_2, \dots, x_n надходять на вхідний шар (який не вважається шаром нейронної мережі), а потім сигнали розподіляються на вхідний шар звичайних нейронів. На кожному ребрі від нейрона вхідного шару до нейрона вихідного шару написано число – вага відповідного зв'язку.

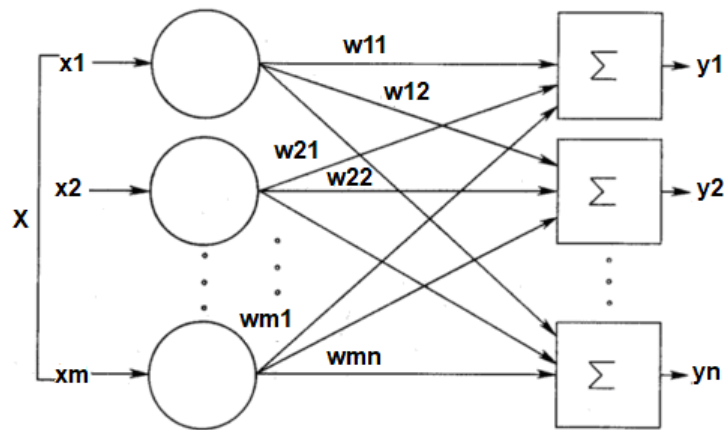


Рисунок 1.6 – Схема одношарової нейронної мережі

Багатошарова нейронна мережа (англ. Multilayer neural network) - нейронна мережа, що складається з вхідного, вихідного і розташованого(их) між ними одного (декілька) прихованих шарів нейронів (рис.1.7) [7].

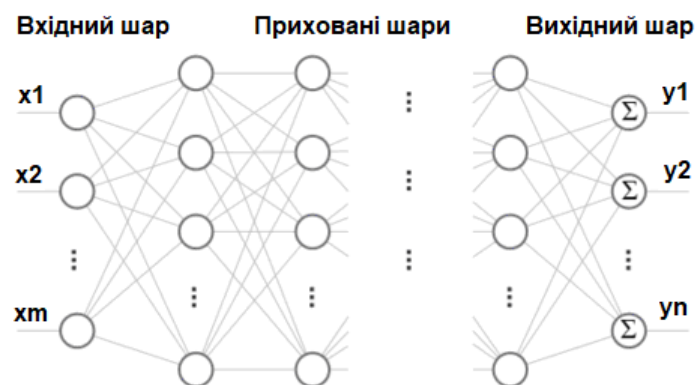


Рисунок 1.7 – Схема багатошарової нейронної мережі

Крім вхідного та вихідного шарів, ці нейронні мережі містять проміжні, приховані шари. Такі мережі мають набагато більші можливості, ніж одношарові нейронні мережі, проте методи навчання нейронів прихованого шару були розроблені відносно недавно.

Роботу прихованих шарів нейронів можна порівняти із роботою великого заводу. Продукт (вихідний сигнал) на заводі збирається на стадіях на верстатах. Після кожного верстата виходить якийсь проміжний результат. Приховані шари також перетворюють вхідні сигнали деякі проміжні результати.

Мережі прямого поширення (Feedforward neural network) (feedforward мережі) – штучні нейронні мережі, в яких сигнал поширюється суворо від вхідного шару до вихідного. У зворотному напрямку сигнал не поширюється.

Усі мережі, описані вище, були мережами прямого поширення, як впливає з визначення. Такі мережі широко використовуються і успішно вирішують певний клас завдань: прогнозування, кластеризація і розпізнавання.

Однак сигнал у нейронних мережах може йти і у зворотний бік.

Структура мережі із зворотними зв'язками показана на рис.1.8 [7].

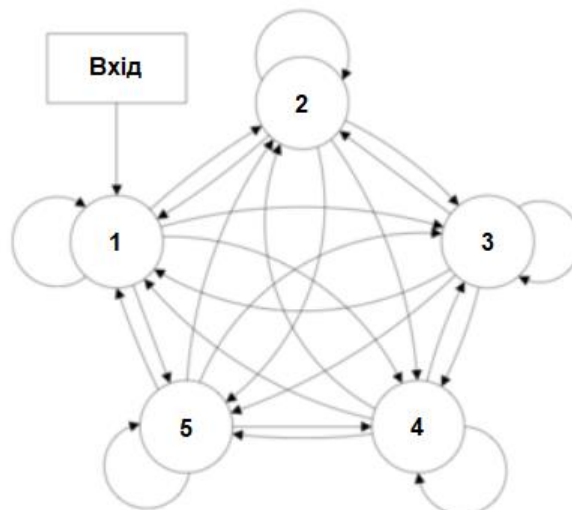


Рисунок 1.8 – Схема мережі із зворотними зв'язками

Мережі із зворотними зв'язками (Recurrent neural network) – штучні нейронні мережі, у яких вихід нейрона може знову подаватися з його вхід. У

загальному випадку це означає можливість поширення сигналу від виходів до входів.

У мережах прямого поширення вихід мережі визначається вхідним сигналом та ваговими коефіцієнтами при штучних нейронах. У мережах із зворотними зв'язками виходи нейронів можуть повертатися на входи. Це означає, що вихід якогось нейрона визначається не тільки його вагами та вхідним сигналом, але ще й попередніми виходами (оскільки вони знову повернулися на входи).

1.4 Навчання нейронних мереж

Навчання нейронної мережі – пошук такого набору вагових коефіцієнтів, у якому вхідний сигнал після проходження мережею перетворюється на потрібний нам вихідний [8].

Це визначення «навчання нейронної мережі» відповідає і біологічним нейромережам. Наш мозок складається з величезної кількості пов'язаних один з одним нейромереж, кожна з яких окремо складається з нейронів одного типу (з однаковою функцією активації). Наш мозок навчається завдяки зміні синапсів – елементів, які посилюють чи послаблюють вхідний сигнал.

Якщо навчати мережу, використовуючи лише один вхідний сигнал, то мережа просто «запам'ятає правильну відповідь», а як тільки ми подамо трохи змінений сигнал, замість правильної відповіді отримаємо нісенітницю. Ми чекаємо від мережі здатності узагальнювати якісь ознаки та вирішувати завдання на різних вхідних даних. Саме з цією метою і створюються навчальні вибірки.

Навчальна вибірка – кінцевий набір вхідних сигналів (іноді разом із правильними вихідними сигналами), якими відбувається навчання мережі [8].

Після навчання мережі, тобто коли мережа видає коректні результати всім вхідних сигналів з навчальної вибірки, її можна використовувати практично. Однак перш ніж відразу використовувати нейронну мережу, зазвичай оцінюють якість її роботи на так званій тестовій вибірці.

Тестова вибірка – кінцевий набір вхідних сигналів (іноді разом із правильними вихідними сигналами), якими відбувається оцінка якості роботи мережі.

Саме навчання нейронної мережі можна розділити на два підходи: навчання з учителем [9] та навчання без вчителя [9]. У першому випадку ваги змінюються так, щоб відповіді мережі мінімально відрізнялися від готових правильних відповідей, а в другому мережа самостійно класифікує вхідні сигнали.

1.5 Завдання і сфери застосування нейромереж

Різні варіанти нейромереж створюються для вирішення декількох типів різних завдань (рис.1.9) [10]:

- класифікація – віднесення об'єктів до потрібного класу;
- регресія – прогнозування результату у вигляді чисел;
- розпізнавання – виділення об'єкта серед величезної множини інших схожих (приклад – мережа може виділити конкретну особу в натовпі);
- кластеризація – поділ об'єктів на кілька груп за якоюсь ознакою, невідомою раніше. Це, наприклад, розбивка документів різні області;
- генерація – народження чогось нового у межах заданої тематики;
- прогнозування – на основі отриманих даних штучний інтелект формулює прогнози на задану тему на певний час.

Залежно від завдання, яке можуть вирішувати штучні нейронні мережі (вона у кожного своя), вони використовуються у різних галузях. Перерахуємо сфери, де вони найбільш популярні:



Рисунок 1.9 – Завдання для нейромереж

– медицина. Штучний інтелект допомагає обробляти знімки та інші дані досліджень і цим дозволяє лікарям встановлювати точний діагноз, при цьому витратити менше часу;

– освіта. Викладачі за допомогою штучних мереж мають можливість швидше перевіряти домашні завдання, за короткий час скласти складні презентації та плани уроків;

– мистецтво. Нейромережі створюють зображення, твори літератури та музику;

– будівництво та архітектура. Штучний інтелект корисний забудовникам, щоб вибрати матеріали, прогнозувати час виконання робіт;

– безпека. Нейромережі мають можливість розпізнавати звичайні особи та шляхом стеження у громадських місцях обчислювати злочинців, які перебувають у розшуку;

– банківська сфера. Нейронна мережа аналізує кредитну історію клієнтів, створює прогнози біржових індексів;

– виробництво. Штучний інтелект бере участь у відстеженні виробничих процесів, дає змогу контролювати продукцію на підприємствах.

1.6 Критерії якості функціонування нейронних мереж

Головне, що необхідно знати для коректної реалізації процесу навчання нейромережі – критерії якості.

У класичному варіанті алгоритму back propagation критерієм є мінімум середніх квадратів помилок між необхідними вихідними значеннями нейромережі і реальними [11]:

$$E = \frac{1}{N} \sum_{i=1}^N (d_i - y_i)^2, \quad (1.6)$$

тут $\{d_i\}$ – необхідні відгуки кожного i -го вхідного вектора;

$\{x_i\}$ – отримуваний відгук мережі, при пред'явленні i -го вектора спостережень.

Тобто задачею є підбір вагових коефіцієнтів зв'язків між нейронами так, щоб мінімізувати цей функціонал.

Але, як показала практика, критерій (1.6) підходить далеко не для всіх завдань, де застосовуються нейромережі і методом проб і помилок були знайдені інші, що призводять до кращих результатів.

Сьогодні не існує точних рекомендацій для побудови структури і критеріїв якості навчання нейронних мереж. Побудова та навчання нейромережі і сьогодні лишається дуже суб'єктивним і творчим процесом. Тут допомагає лише досвід та базові знання про те, як усе це працює.

Спочатку розглянемо критерії для завдань регресії, де вихідне значення нейрона представлено в числовій шкалі (ціни, зростання, вага, кількість тощо). Тут перший критерій – це класичний середній квадрат помилок (1.6). Його недолік у тому, що він утворює багато локальних мінімумів і найчастіше навчання у них і «застряє».

З іншого боку, критерій (1.6) різко зростає у разі великих помилок, оскільки помилка зводиться у квадрат. Значні помилки, як правило,

зустрічаються рідко, але якщо вже з'явилися, це сильно позначиться на критерії якості.

Як варіант, можна скористатися середньою сумою модулів помилок (mean absolute error) [11]:

$$E = \frac{1}{N} \sum_{i=1}^N |d_i - y_i|. \quad (1.7)$$

У цьому випадку великі, рідкісні помилки вже не так сильно впливають на величину E . Можна зазначити, що функція модуля не має похідної в нижній точці. Але це вже не проблема, чисельно її можна визначити з деякими припущеннями. Тому критерій якості (1.7) теж застосовується практично.

Наступний критерій якості – середній квадрат логарифмічних помилок (mean squared logarithmic error) [11]:

$$E = \frac{1}{N} \sum_{i=1}^N (\log(d_i + 1) - \log(y_i + 1))^2. \quad (1.8)$$

На відміну від звичайної суми квадратів, використання логарифмів значно зменшує рідкісні помилки і вони вже не так сильно впливають на величину E . Це ще один спосіб боротьби з рідкісними, але значущими помилками на виході нейромережі.

Для завдань прогнозування виконують нормування суми модулів і використовують критерій під назвою «середній абсолютний відсоток помилок» (mean absolute percentage error) [11]:

$$E = \frac{1}{N} \sum_{i=1}^N \frac{|d_i - y_i|}{\max(y_i, \epsilon)} \cdot 100. \quad (1.9)$$

Наприклад, нейромережа видає абсолютну помилку 1000. Це багато чи мало? Тут важливо знати відносно яких величин виходить 1000. Якщо ми оперуємо величинами 10 млн., то 1000 це помилка в частки відсотка. А якщо використовуються значення порядку 10 тис, то 1000 – це помилка в 10%. Щоб уникнути абсолютних величин і перевести помилку у відносні – відсотки, якраз і запровадили такий критерій якості.

Наступні критерії застосовуються у завданнях класифікації (розпізнавання). Припустимо, що нейромережа має кілька виходів M , на кожному з яких з'являються значення $+1$ або -1 , тобто [11]:

$$y_i \in \{-1, 1\}. \quad (1.10)$$

Тут як критерій якості можна скористатися мірою hinge або square hinge (хіндж) [11]:

$$E = \frac{1}{N} \sum_{i=1}^N \max(1 - d_i \cdot y_i, 0), \quad (1.11)$$

$$E = \frac{1}{N} \sum_{i=1}^N (\max(1 - d_i \cdot y_i, 0))^2. \quad (1.12)$$

Він працює очевидним чином. Якщо нейромережа правильно видає значення, тобто

$$d_i = y_i, \quad (1.13)$$

то

$$\max(1 - d_i \cdot y_i, 0) = 0, \quad (1.13)$$

інакше, на виході отримуватимемо [11]:

$$\max(1 - d_i \cdot y_i, 0) = 2. \quad (1.14)$$

Фактично ми тут збільшуємо параметр E при кожному розбіжності виходу із заданим значенням.

На сьогоднішній день у задачах класифікації образів при двох класах (наприклад, класифікація на чоловіків і жінок, собак і кішок тощо) широко використовується бінарна крос-ентропія (binary crossentropy) [12]:

$$E = -\frac{1}{N} \sum_{i=1}^N d_i (\log y_i + (1 - d_i) \log(1 - y_i)). \quad (1.15)$$

При цьому необхідні виходи відзначаються бінарними значеннями: $d_i \in \{-1,1\}$ або $d_i \in \{0,1\}$. Проте, за більшої кількості класів $k > 2$ слід вибирати категоріальну крос-ентропію (categorical crossentropy).

Нарешті вважається, що при обробці тексту добре підходить критерій логарифмічний гіперболічний косинус (logcosh) [13]:

$$E = \frac{1}{N} \sum_{i=1}^N \log(\cosh(y_i - d_i)). \quad (1.16)$$

Величина $\log(\cosh(x))$ близька до $x^2/2$ при малих x і приблизно дорівнює $|x| - \log 2$ при великих x .

Це лише деякі часто використовуються на практиці критерії якості роботи неймережі. Постійно з'являються нові й нові варіанти. Можливо, ви будете одним із тих, хто придумає черговий популярний захід і про нього почнуть говорити на курсах навчання неймережі.

На цьому ми завершимо загальний теоретичний огляд основних моментів для підготовки неймережі до процесу навчання і надалі використовуватимемо ці знання при вирішенні конкретних практичних завдань.

1.7 Висновки по розділу 1

Перші нейронні мережі у вигляді комп'ютерного коду були реалізовані ще в 1950-х роках. Тоді вони вирішували прості задачі класифікації, на більш складні завдання не вистачало обчислювальної потужності. Новий поштовх нейромережі отримали в 2010-х роках, коли почали вирішувати задачі аналізу великих об'ємів даних і прогнозування, розпізнавання та синтезу тексту або мови, комп'ютерного зору і багатьох інших задач.

Не оминули нейромережі і мультимедіа технології. Сьогодні багато програмних продуктів включають спеціальні модулі зі штучним інтелектом, що дозволяють виконувати не тільки обробку звукових, графічних та відеоданих за допомогою нейромереж, але й суттєво модифікувати мультимедійний контент, прибираючи або добавляючи окремі об'єкти на зображенні, або сильно видозмінюючи головний об'єкт на фото. Причому результати такої обробки при правильному підході виглядають дуже реалістичними. Вершиною складності є генерація нейромережею власного графічного контенту на основі.

Отже, штучний інтелект можна розглядати як повноцінний інструмент створення мультимедійного контенту, поряд з іншими, класичними методами обробки. Як показує практика, сьогодні не існує точних рекомендацій для побудови структури, процесу навчання і критеріїв якості нейронних мереж. Робота з нейромережею лишається дуже суб'єктивним і творчим процесом. Тут допомагає досвід та базові знання про функціонування даного інструменту.

Метою кваліфікаційної роботи є теоретичний аналіз основ побудови, функціонування, навчання та оцінки якості навчання нейронних мереж для обробки і генерації фотореалістичних зображень, а також практичне дослідження особливостей роботи і результатів, що дають на сьогоднішній день такі нейромережі.

Дана кваліфікаційна робота магістра виконана на кафедрі Медіаінженерії та інформаційних радіоелектронних систем ХНУРЕ. На кафедрі МІРЕС проводяться дослідження в таких наукових областях, як виявлення та розпізнавання БПЛА за результатами відеоспостереження [14-19], систем технічного зору роботів [20, 21]. Цілий ряд студентських доповідей [22-25] присвячені тематиці обробки зображень. Отже, тематика даної кваліфікаційної роботи відповідає традиційному науковому напрямку кафедри МІРЕС.

2 ПРИНЦИПИ РОБОТИ НЕЙРОМЕРЕЖ ДЛЯ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ

2.1 Основи генерації зображень

У більшості сучасних додатків глибокого навчання використовується нейронний дискримінантний тип (дискримінатор), а SPADE (SPatially-Adaptive (DE)normalization) – це генеративна нейронна мережа (генератор) [26].

2.1.1 Дискримінатори зображення

Дискримінатор займається класифікацією даних, що вводяться. Наприклад, класифікатор зображення – це дискримінатор, який бере зображення і вибирає одну відповідну мітку класу, наприклад, визначає зображення як "собаку", "автомобіль" або "світлофор", тобто вибирає мітку, яка повністю описує зображення. Отриманий класифікатором висновок зазвичай представлений у вигляді вектора чисел $\{v_i\}$, де v_i це число від 0 до 1, що виражає впевненість мережі у приналежності зображення до вибраного i -го класу (рис.2.1).

Дискримінатор також може скласти список класифікацій. Він може класифікувати кожен піксель зображення як клас «людей» або «машин» (так звана «семантична сегментація»).

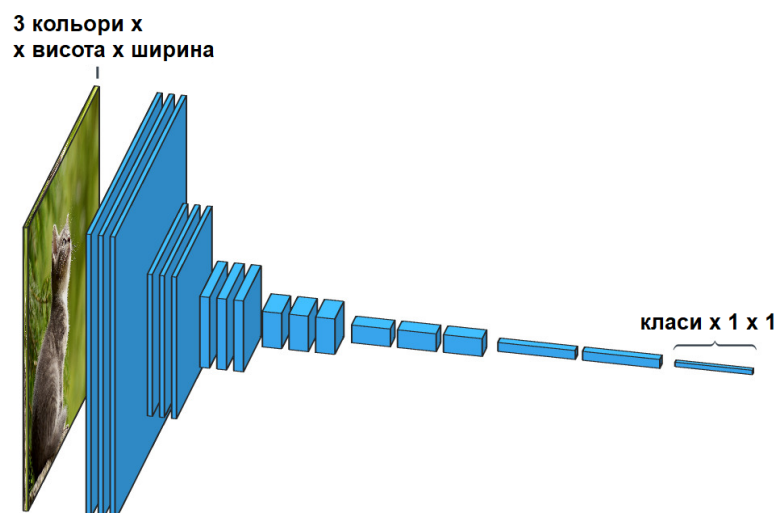


Рисунок 2.1 – Вирішення задачі класифікації

Класифікатор бере зображення з 3 каналами (червоний, зелений та синій) та зіставляє його з вектором впевненості у кожному можливому класі, який може представляти зображення.

Оскільки зв'язок між зображенням та його класом дуже складна, нейронні мережі пропускають його через стек із безлічі шарів, кожен із яких «трохи» обробляє його і передає свій висновок на наступний рівень інтерпретації.

2.1.2 Генератори зображення

Генеративна мережа на зразок SPADE отримує набір даних і прагне створювати нові оригінальні дані, які виглядають так, начебто вони належать до цього класу даних. При цьому дані можуть бути будь-якими: звуками, мовою чи ще чимось, але ми зосередимося на зображеннях. У загальному випадку введення даних в таку мережу є просто вектором випадкових чисел, причому кожен з можливих наборів даних, що вводяться, створює своє зображення (рис.2.2).

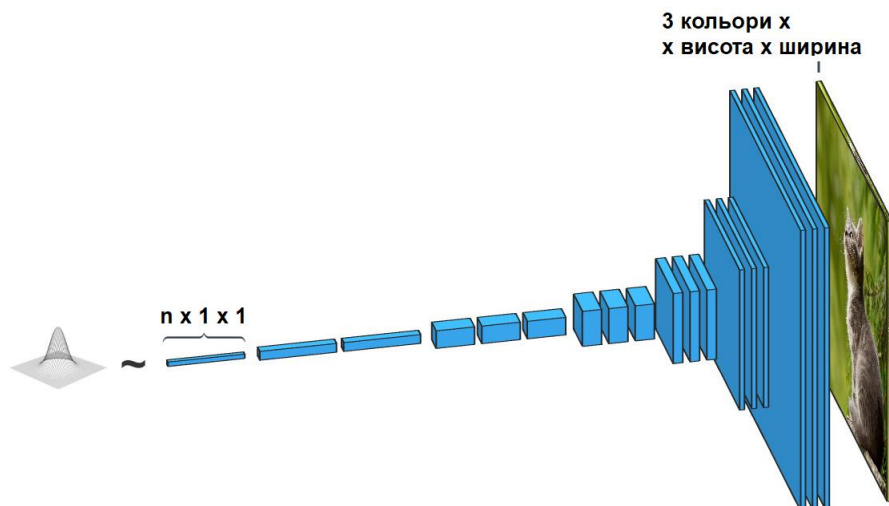


Рисунок 2.2 – Вирішення задачі генерації

Генератор на основі випадкового вхідного вектора працює фактично протилежно до класифікатора зображення. У генераторах «умовних класів» вхідний вектор – це, власне, вектор цілого класу даних.

Як зрозуміло, SPADE використовує набагато більше, ніж просто «випадковий вектор». Система керується своєрідним кресленням, яке називається «картою сегментації». Остання показує, що і де розміщувати. SPADE проводить процес, обернений до семантичної сегментації, згаданої нами вище. Загалом, дискримінаційна задача, що переводить один тип даних в інший, має аналогічне завдання, але йде іншим, незвичним шляхом.

Сучасні генератори та дискримінатори зазвичай використовують згорткові мережі для обробки своїх даних.

Між класифікатором та генератором зображення є одна важлива відмінність, і полягає воно в тому, як саме вони змінюють розмір зображення під час його обробки. Класифікатор зображення повинен зменшувати його, поки зображення не втратить всю просторову інформацію і залишаться лише класи. Це може бути досягнуто об'єднанням шарів, або через використання згорткових мереж, через які пропускають окремі пікселі. Генератор створює зображення за допомогою процесу, зворотної «згортки», який називається згортковим транспонуванням. Його ще часто плутають із «деконволюцією» чи «зворотною згорткою».

На рис.2.3, а показана ситуація, коли звичайна згортка 2×2 з кроком «2» перетворює кожен блок 2×2 на одну точку, зменшуючи вихідні розміри на $1/2$. На рис.2.3, б показана зворотна ситуація, транспонована згортка 2×2 з кроком «2» генерує блок 2×2 з кожної точки, збільшуючи вихідні розміри в 2 рази.

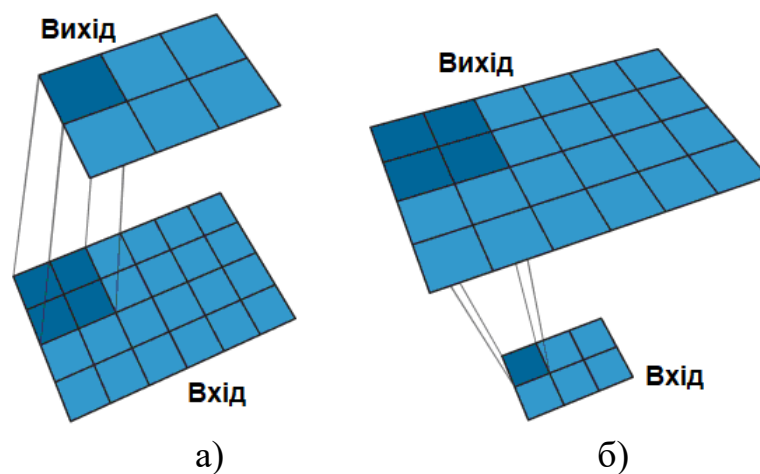


Рисунок 2.3 – Звичайна (а) і транспонована згортка (б)

2.2 Тренування генератора

Теоретично, згорткова нейронна мережа може генерувати зображення описаним вище чином. Але як її натренувати? Тобто, якщо враховувати набір вхідних даних-зображень, як ми можемо налаштувати параметри генератора (у нашому випадку SPADE), щоб він створював нові зображення, які виглядають так, наче вони відповідають запропонованому набору даних.

Для цього потрібно проводити порівняння з класифікаторами зображень, де кожне має правильну мітку класу. Знаючи вектор передбачення мережі та правильний клас, ми можемо використовувати алгоритм зворотного розповсюдження (backpropagation algorithm), щоб визначити параметри оновлення мережі (рис.2.4). Це необхідно, щоб підвищити її точність у визначенні потрібного класу і зменшити вплив інших класів.

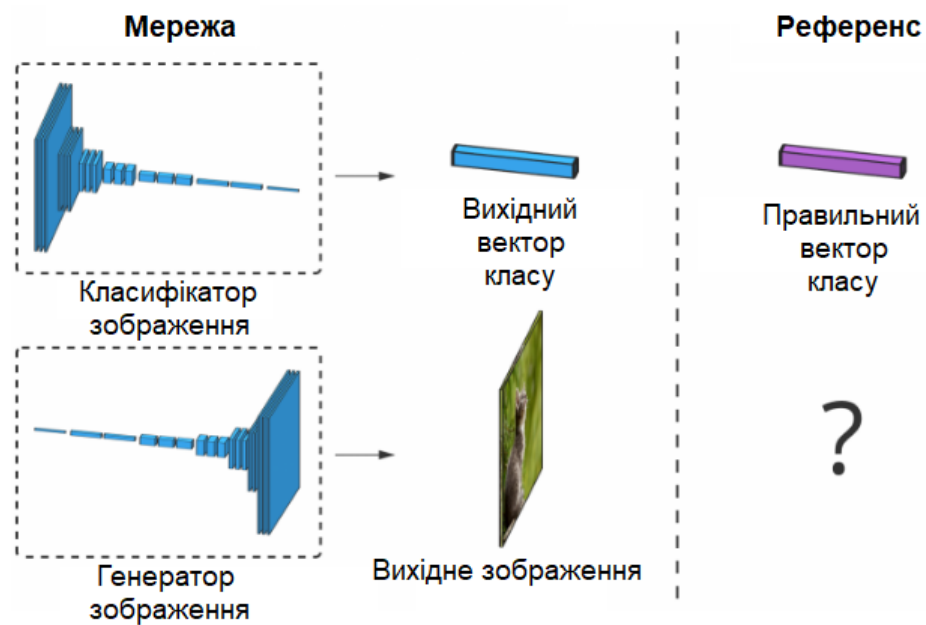


Рисунок 2.4 – Процес тренування генератора

Точність класифікатора зображення можна оцінити, поелементно порівнявши його висновок із правильним вектором класу. Але для генераторів немає «правильного» вихідного зображення.

Проблема в тому, що коли генератор створює зображення, для кожного пікселя немає «правильних» значень (ми не можемо порівняти результат, як у випадку класифікатора за попередньо підготовленою базою, — прим. пров.). Теоретично будь-яке зображення, яке виглядає правдоподібно і схоже на цільові дані, є дійсним, навіть якщо його значення пікселів сильно відрізняються від реальних зображень.

Отже, як ми можемо повідомити генератору, у яких пікселях він повинен змінювати свій висновок і яким чином він може створювати більш реалістичні зображення (тобто як подавати сигнал про помилку)? Дослідники багато міркували з цього питання, і насправді це досить складно. Більшість ідей, таких як обчислення деякої середньої відстані до реальних зображень, дають розмиті картинки низької якості.

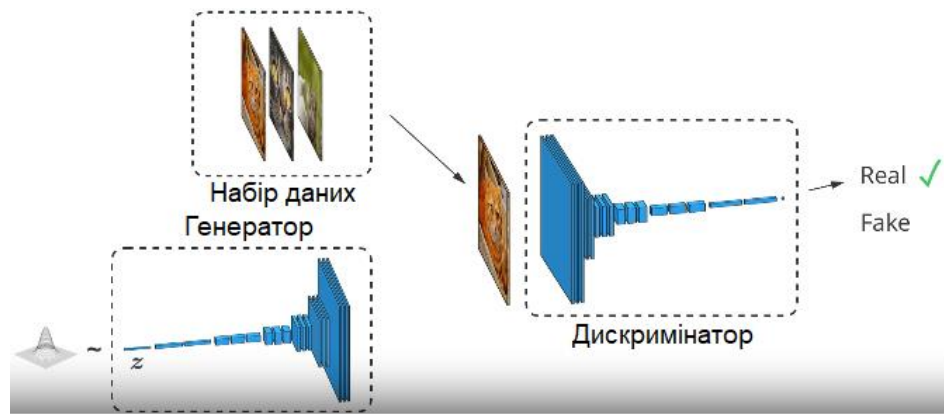
В ідеалі ми могли б «виміряти», наскільки реалістично виглядають згенеровані зображення через використання концепції «високого рівня», як «Наскільки складно це зображення відрізнити від реального?»...

2.3 Генеративні змагальні мережі

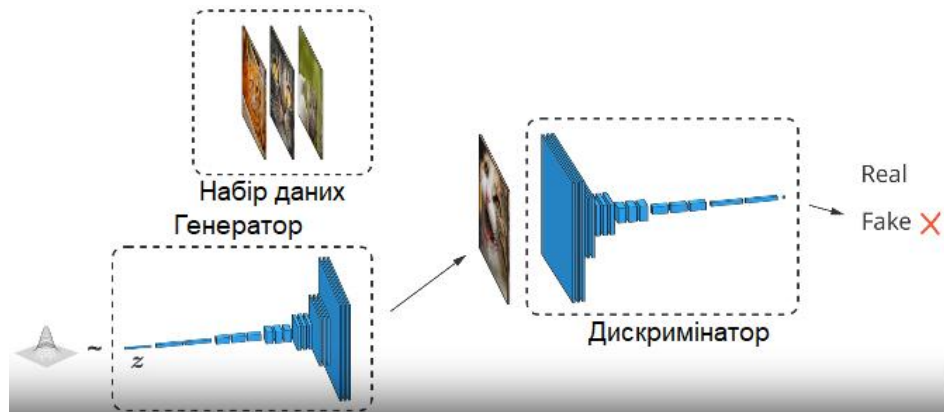
Саме це реалізувано в рамках роботи [27]. Ідея полягає в тому, щоб генерувати зображення, використовуючи дві нейронні мережі замість однієї (рис. 2.5):

- одна мережа – генератор (рис. 2.5, б),
- друга – класифікатор зображень (дискримінатор рис. 2.5, а).

Завдання дискримінатора у тому, щоб відрізнити вихідні зображення генератора від реальних зображень з первинного набору даних (класи цих зображень позначені як «підроблені» і «реальні»). Робота ж генератора полягає в тому, щоб обдурити дискримінатор, створюючи зображення максимально схожі на зображення в наборі даних. Можна впевнено сказати, що генератор і дискримінатор є противниками у процесі. Звідси і походить назва: генеративно-змагальна мережа.



а)



б)

Рисунок 2.5 – Генеративно-змагальні мережі
(а – класифікатор зображень, б – генератор зображень)

Генеративно-змагальна мережа, що працює на базі випадкового векторного введення. У цьому прикладі один із виходів генератора намагається обдурити дискриміратор при виборі «реального» зображення.

Тепер ми можемо використовувати повідомлення про помилку, засноване виключно на передбаченні дискримінатора: значення від 0 (фальшивий) до 1 (реальний). Оскільки дискриміратор є нейронною мережею, ми можемо ділитися його висновками про помилки та з генератором зображень. Тобто дискриміратор може повідомляти генератору, де і як той повинен коригувати свої зображення, щоб краще «обдурити» дискриміратор (тобто як підвищити реалістичність своїх зображень).

2.4 Алгоритм роботи генеративно-змагальної мережі GAN

Розглянутий нами GAN у своїй роботі слідує описаній вище логіці. Його дискримінатор D аналізує зображення x і отримує значення $D(x)$ від 0 до 1, яке відображає його ступінь впевненості в тому, реальне це зображення, або підроблене генератором. Його генератор G отримує випадковий вектор нормально розподілених чисел Z і виводить зображення $G(z)$, яке може «обдурити» дискримінатор (за фактом, це зображення $D(G(z))$).

Одне з питань – як навчають GAN і яку функцію втрат використовують розробники для вимірювання продуктивності мережі. У загальному випадку функція втрат повинна зростати в міру навчання дискримінатора та знижуватися в міру навчання генератора. Функція втрат вихідного GAN використовувала два наступні параметри:

- перший – є ступенем того, як часто дискримінатор правильно класифікує реальні зображення як реальні;
- другий – наскільки добре дискримінатор виявляє підроблені зображення:

$$L_{GAN}(D, G) = E_{\bar{x} \sim p_{data}}[\log D(\bar{x})] + E_{\bar{z} \sim N}[\log(1 - D(G(\bar{z})))] \quad (2.1)$$

У процесі навчання знаходження підроблених зображень дискримінатор дає генератору все кращу і кращу зворотний зв'язок у тому, як останній може поліпшити своєї роботи. Таким чином, дискримінатор виконує "learn a loss"-функцію для генератора.

Дискримінатор D виводить своє твердження про те, що зображення є реальним. Це сенс, оскільки $\log D(\bar{x})$ зростає, коли дискримінатор вважає x реальним. Коли ж дискримінатор краще виявляє підроблені зображення, збільшується значення виразу $\log(1 - D(G(\bar{z}))$ (починає прагнути до 1), оскільки $D(G(\bar{z}))$ буде прагнути до 0.

Насправді ми оцінюємо точність, використовуючи цілі партії зображень. Ми беремо багато (але далеко не всі) реальних зображень x і багато випадкових векторів Z , щоб отримати усереднені числа за формулою вище. Потім ми вибираємо поширені помилки та набір даних.

Згодом це призводить до цікавих результатів (рис.2.6).



Рисунок 2.6 – Goodfellow GAN, що імітує набори даних MNIST, TFD та CIFAR-10. Контурні зображення (а) є найближчими в наборі даних до суміжних підробок (б).

Дані результати отримані 9 років тому [27]. На щастя, як свідчить SPADE та інші мережі, машинне навчання продовжує швидко прогресувати.

2.5 Проблеми тренування

Генеративно-змагальні мережі сумно відомі своєю складністю у підготовці та нестабільністю роботи. Одна з проблем полягає в тому, що якщо генератор дуже сильно випереджає дискримінатор в темпах навчання, то його вибірка зображень звужується до тих, які допомагають йому обдурити дискримінатор.

Фактично, в результаті навчання генератора зводиться до створення єдиного універсального зображення для обману дискримінатора. Ця проблема називається «режимом колапсу» (рис.2.7) [28].



Рисунок 2.7 – Режим колапсу GAN

В режимі колапсу GAN багато з зображень виглядають дуже схожими один на одного.

Інша проблема полягає в тому, що коли генератор ефективно обманює дискримінатор $D(G(\bar{z}))$ він оперує дуже маленьким градієнтом, тому $L_{GAN}(D, G)$ не може отримати достатньо даних для знаходження справжньої відповіді, при яких це зображення виглядало б більш реалістичним.

Зусилля дослідників щодо вирішення цих проблем в основному були спрямовані на зміну структури функції втрат. Одним із простих змін, запропонованих у [Xudong Mao et al., 2016], є заміна функції втрат L_{GAN} на пару простих функцій V_{LSGAN} в основі яких лежать квадрати меншої площі. Це призводить до стабілізації процесу тренування, отримання більш якісних зображень та меншого шансу колапсу з використанням незагасаючих градієнтів.

Інша проблема, з якою зіткнулися дослідники – це складність отримання зображень з високою роздільною здатністю, частково через те, що детальніше зображення дає дискримінатору більше інформації виявлення підроблених зображень. Сучасні GAN починають навчати мережу з картинок у низькій роздільній здатності і поступово додають все більше і більше шарів, поки не буде досягнутий бажаний розмір зображення (рис.2.8).

Поступове додавання шарів з більш високою роздільною здатністю під час навчання GAN істотно підвищує стабільність всього процесу, а також швидкість та якість одержуваного зображення.

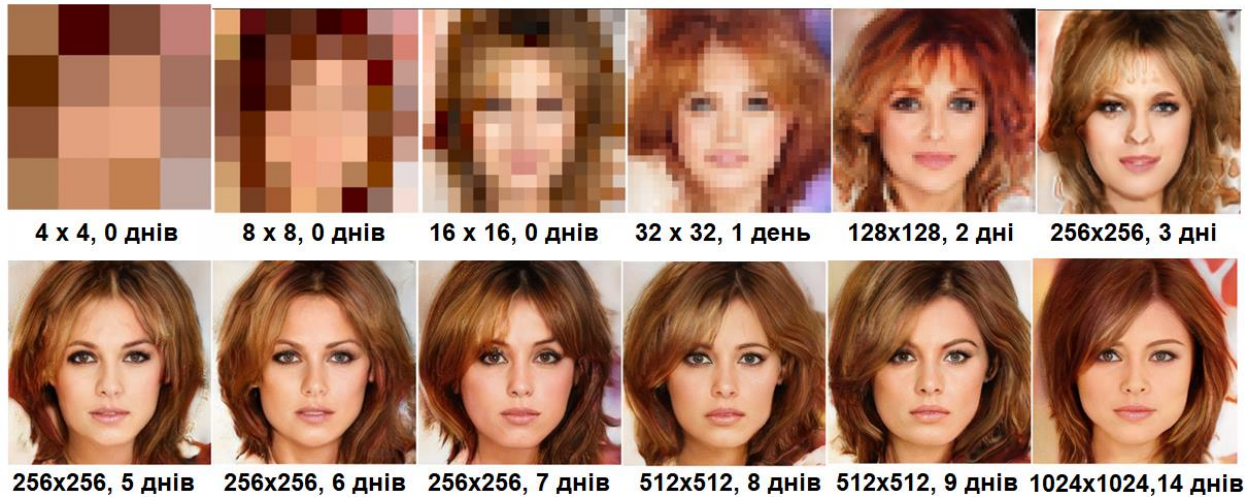


Рисунок 2.8 – Збільшення розрізнення в процесі навчання GAN

2.6 Image-to-image трансляція

Досі ми говорили про те, як генерувати зображення з випадкових наборів вступних даних. Але SPADE не просто використовує довільні дані. Ця мережа використовує зображення, яке називається картою сегментації: вона призначає кожному пікселю клас матеріалу (наприклад трава, дерево, вода, камінь, небо). З цього зображення-карти SPADE і генерує те, що виглядає як фотографія. Цей процес називається "Image-to-image трансляція" (рис.2.9).



Рисунок 2.9 – Шість різних видів трансляції Image-to-image, продемонстровані мережею pix2pix

Для того, щоб генератор вивчив цей підхід, йому потрібен набір карт сегментації та відповідні фотографії. Модифікуємо архітектуру GAN, щоб і генератор, і дискримінатор отримали карту сегментації. Генератору, звісно, потрібна карта, щоб знати, «у який бік малювати». Дискримінатору вона також потрібна, щоб переконатися, що генератор розміщує правильні речі в правильних місцях (рис.2.10).

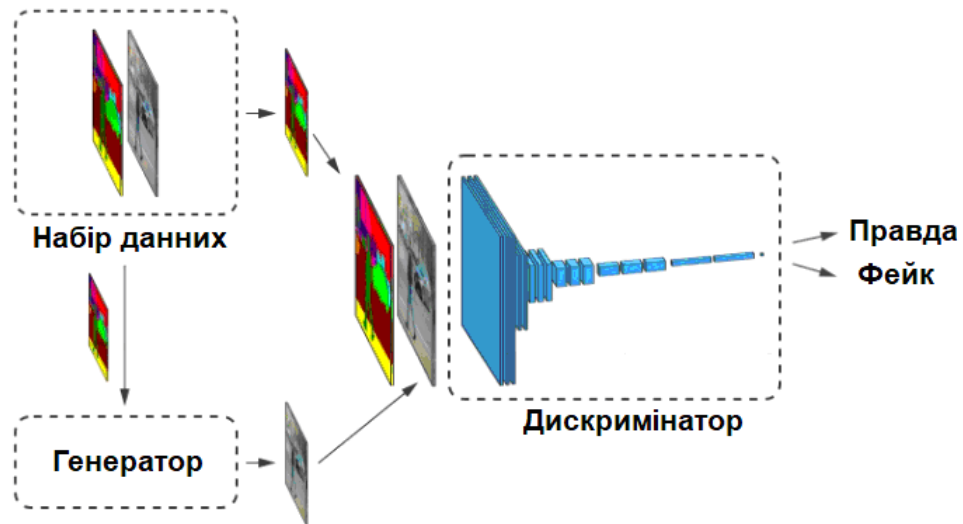


Рисунок 2.10 – Модифікація архітектури GAN, щоб генератор і дискримінатор отримали карту сегментації

У ході навчання генератор вчиться не ставити траву там, де на карті сегментації зазначено «небо», оскільки інакше дискримінатор легко визначить підроблене зображення, і таке інше.

Для image-to-image трансляції вхідне зображення приймається як генератором, і дискримінатором. Дискримінатор додатково отримує або вихідні дані генератора, або дійсний висновок з набору навчальних даних.

2.7 Приклад розробки image-to-image транслятора

Розглянемо реальний image-to-image транслятор [28]. Для image-to-image транслятора наш генератор створює зображення та приймає його як вхідний. Можна було би просто використовувати карту згорткових шарів, але

оскільки згорткові шари поєднують значення тільки в невеликих ділянках, то потрібно надто багато шарів для передачі інформації про зображення у високій роздільній здатності.

Pix2pixHD вирішує цю проблему більш ефективно за допомогою «Кодувальника», який зменшує масштаб вхідного зображення, за яким слідує «Декодер», що збільшує масштаб для отримання вихідного зображення (рис.2.11).

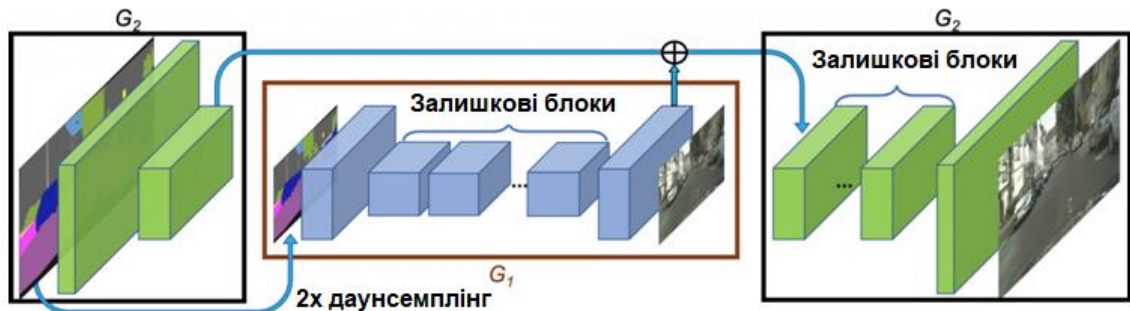


Рисунок 2.11 – Схема мережі pix2pixHD на високому рівні.

Залишкові блоки та операція "+" відносяться до технології "skip connections" з Residual neural network.

Майже всі сучасні згорткові нейронні мережі використовують пакетну нормалізацію або один із її аналогів, щоб прискорити та стабілізувати процес тренування. Активація кожного каналу зсуває середнє до 0 і стандартне відхилення до 1 до того, як пара параметрів каналів β і γ дозволяють їм знову денормалізуватися:

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} \cdot \gamma + \beta, \quad (2.2)$$

де $E[x]$ – математичне сподівання, середня величина x ;

$Var[x]$ – змінна складова x .

На жаль, пакетна нормалізація шкодить генераторам, ускладнюючи мережі реалізацію деяких типів обробки зображень. Замість нормалізувати

партію зображень, pix2pixHD використовує стандарт нормалізації, який нормалізує кожне зображення окремо.

2.8 Навчання pix2pixHD

Сучасні GAN, такі як pix2pixHD та SPADE, вимірюють реалістичність своїх вихідних зображень трохи інакше, ніж це було описано для оригінального дизайну генеративно-змагальних мереж.

Для вирішення проблеми генерації зображень у високій роздільній здатності pix2pixHD використовує три дискримінатори однакової структури, кожен з яких отримує вихідне зображення в різному масштабі (звичайний розмір, зменшений у 2 рази та зменшений у 4 рази).

Для визначення своєї функції втрат pix2pixHD використовує V_{LSGAN} , а також включає інший елемент, призначений для того щоб зробити висновки генератора більш реалістичними (незалежно від того, чи допомагає це обдурити дискримінатор). Цей елемент L_{FM} називається «співставлення ознак» — він спонукає генератор зробити розподіл шарів при симуляції дискримінації однаковою між реальними даними та виходами генератора, зводячи до мінімуму $L_1 Distance$ між ними.

Таким чином, оптимізація зводиться до такого:

$$\min_G \left(\lambda \sum_{k=1,2,3} V_{LSGAN}(G, D_k) + \max_{D_1, D_2, D_3} \sum_{k=1,2,3} L_{FM}(G, D_k) \right). \quad (2.3)$$

де втрати сумуються за трьома дискримінаційними факторами та коефіцієнтом $\lambda = 10$, який контролює пріоритет обох елементів.

Хоча дискримінатори зменшують масштаб зображення доти, доки не розберуть зображення повністю, вони зупиняються на «плямах» розміром

70×70 (у відповідних масштабах). Потім вони просто підсумовують всі значення цих плям для всього зображення.

І такий підхід відмінно працює, оскільки функція L_{FM} дбає про те, щоби зображення реалістично виглядало у великій роздільній здатності, і V_{LSGAN} потрібно лише для перевірки дрібних деталей. Такий підхід також має додаткові переваги у вигляді прискорення роботи мережі, зменшення кількості використовуваних параметрів та можливості її використання для генерації зображень будь-якого розміру.

На рис.2.12 показано приклад, коли мережа $pix2pixHD$ генерує із простих нарисів осіб фотореалістичні зображення з відповідними гримасами. Кожен приклад показує реальне зображення з набору даних CelebA зліва, ескіз виразу обличчя цієї знаменитості у вигляді скетчу та створене із цих даних зображення праворуч.



Рисунок 2.12 – Приклад роботи мережі $pix2pixHD$

Результати роботи мережі $pix2pixHD$ вражають, але неймережі можуть досягти більшого. Виявляється, $pix2pixHD$ дуже програє в одному важливому аспекті.

Розглянемо, що робить $pix2pixHD$ із однокласним введенням, скажімо, з картою, на якій всюди трава. Оскільки вхід є просторово однорідним, виходи першого згорткового шару також однакові. Потім нормалізація екземплярів «нормалізує» всі (ідентичні) значення кожного каналу у зображенні і повертає 0 як висновок всім з них. β -параметр може зрушити це

значення від нуля, але залишається ситуація: вихід більше не залежатиме від того, чи була на вході «трава», «небо», «вода» чи щось інше.

Вирішення даної проблеми міститься в нейромережі SPADE.

2.9 Нейромережа SPADE

Нейромережі досягли принципово нового рівня у створенні зображень із карт сегментації: просторово-адаптивна (де)нормалізація (SPADE).

Ідея SPADE полягає в тому, щоб запобігти втраті семантичної інформації в мережі, дозволяючи карті сегментації керувати параметрами нормалізації γ , а також β локально на рівні кожного окремого шару. Замість того, щоб використовувати тільки одну пару параметрів для кожного каналу, вони обчислюються для кожної просторової точки шляхом подачі карти сегментації зі зниженою дискретизацією через 2 шари згортки.

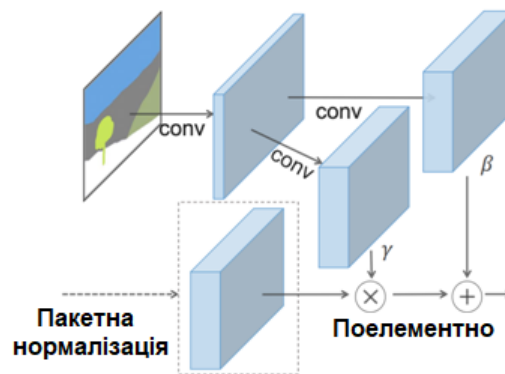


Рисунок 2.13 – Подача карти сегментації зі зниженою дискретизацією через 2 шари згортки

Замість того, щоб розгортати карту сегментації на перший шар, SPADE використовує її версії зі зниженою дискретизацією для модуляції нормалізованих вихідних даних для кожного шару.

Генератор SPADE об'єднує всю цю конструкцію в невеликі «залишкові блоки», які поміщаються між шарами підвищення дискретизації (транспонована згортка, рис.2.14).

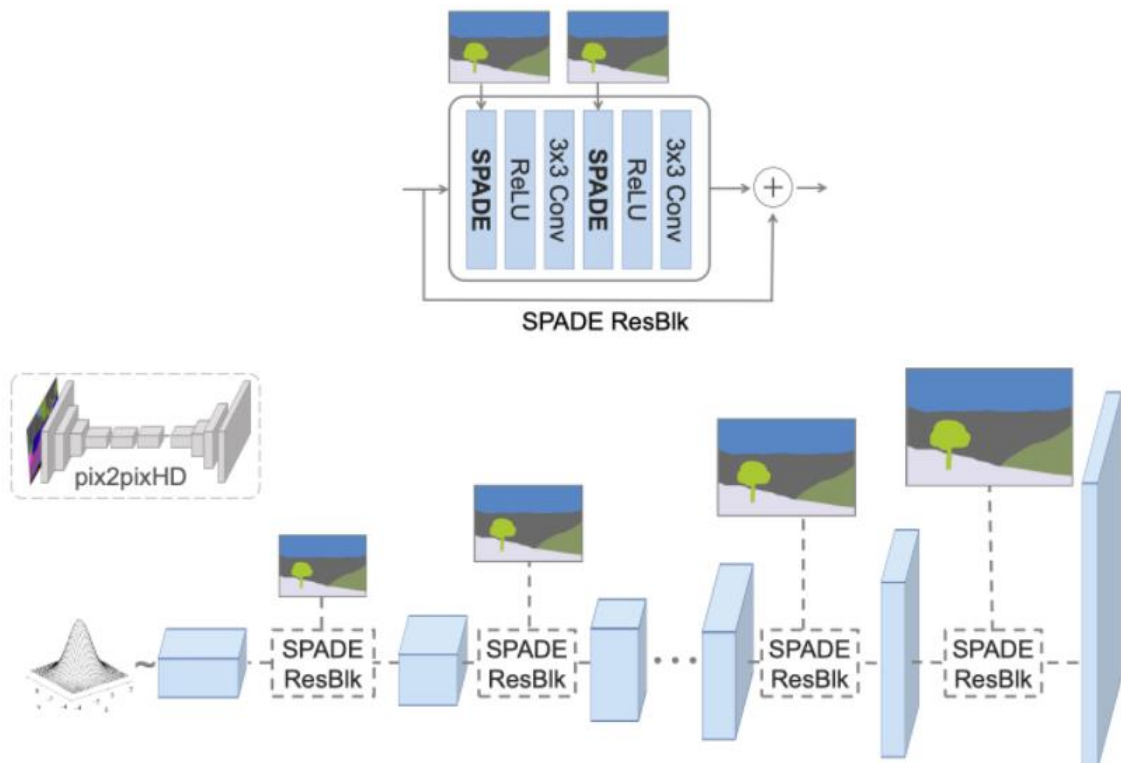


Рисунок 2.14 – Високорівнева схема генератора SPADE у порівнянні з генератором pix2pixHD

Тепер, коли карта сегментації подається «зсередини» мережі, немає необхідності використовувати її як вхідні дані для першого шару. Натомість ми можемо повернутися до вихідної схеми GAN, в якій як вхідні дані використовувався випадковий вектор. Це дає нам додаткову можливість генерувати різні зображення з однієї карти сегментації (мультимодальний синтез). Це також робить непотрібним весь кодировщик pix2pixHD, а це серйозне спрощення.

SPADE використовує ту ж саму функцію втрат, що і pix2pixHD, але з однією зміною: замість зведення в квадрат значень V_{LSGAN} у ній використовується hinge loss.

З цими змінами ми отримуємо гарні результати, показані на рис.2.15. Тут результати нейромережі SPADE порівнюються із результатами нейромережі pix2pixHD, також в першому стовпці показані карти міток (сегментації), а також оригінальні зображення в другому стовпці.

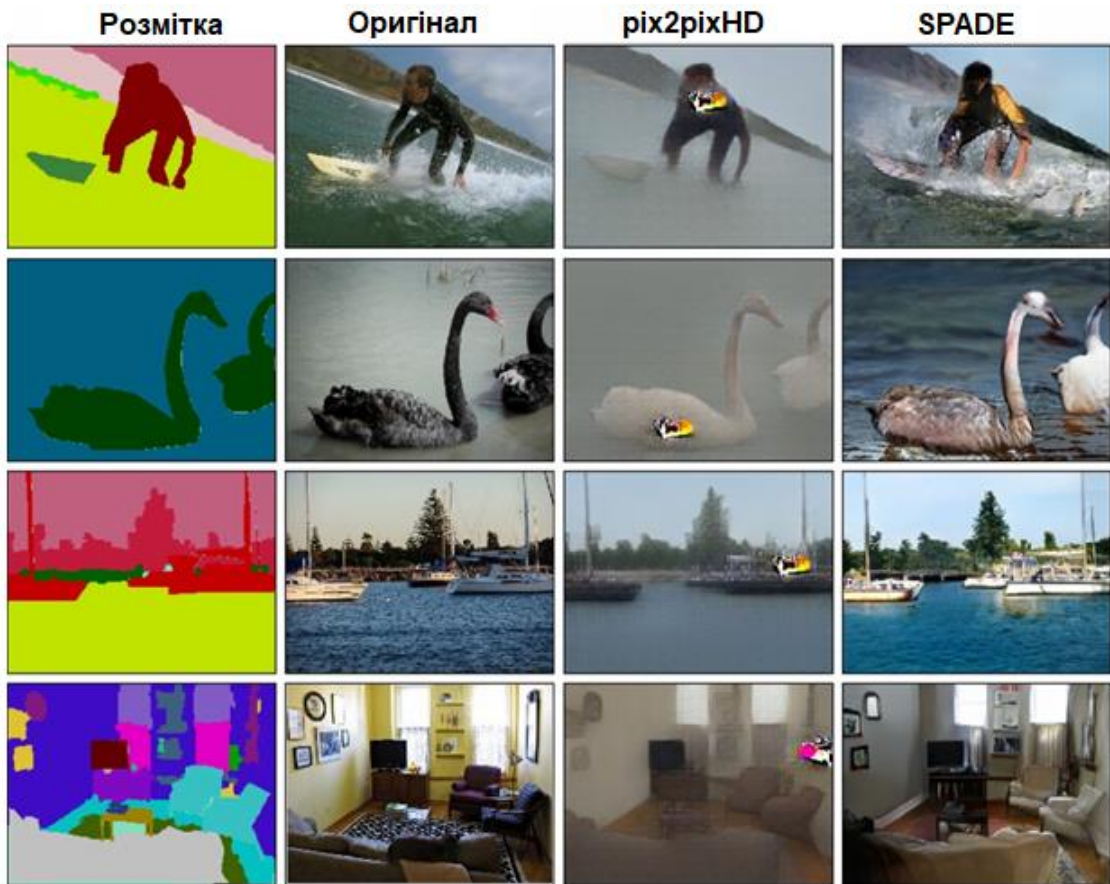


Рисунок 2.15 – Порівняння результатів нейромереж SPADE і pix2pixHD

2.10 Висновки по розділу 2

Зробимо висновок, як SPADE може показувати такі результати. Наприклад, маємо зображення дерева. Алгоритм GauGAN використовує один деревоподібний клас для представлення як стовбура, так і листя дерева. Тим не менш, мережа SPADE дізнається, що вузька частина в нижній частині «дерева» є стовбуром і має бути коричневою, тоді як велика крапля зверху має бути листям.

Сегментація зі зниженою дискретизацією, яку використовує SPADE для модуляції кожного шару, забезпечує подібне «інтуїтивне» розпізнавання.

Стовбур дерева продовжується і в частині крони, яка відноситься до «листя». То як SPADE розуміє, де розмістити там частину ствола, а де листя? Адже, судячи з карти 5x5, там має бути просто «дерево».

Відповідь полягає в тому, що показана ділянка може отримувати інформацію від шарів з нижчою роздільною здатністю, де блок 5x5 містить все дерево цілком. Кожен наступний шар згортки також забезпечує деяке переміщення інформації з зображення, що дає більш повну картину.

SPADE дозволяє карті сегментації безпосередньо модулювати кожен шар, але це не перешкоджає процесу зв'язкового поширення інформації між шарами, як це відбувається, наприклад, у `pix2pixHD`. Це запобігає втраті семантичної інформації, оскільки вона оновлюється у кожному наступному шарі за рахунок попереднього.

3 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ГЕНЕРАЦІЇ ФОТОРЕАЛІСТИЧНИХ ЗОБРАЖЕНЬ НЕЙРОМЕРЕЖАМИ

3.1 Загальні правила складання запитів

Дослідження генерації картинок проведемо на прикладі популярного сервісу Openjourney. Є список правил спілкування, який запропонували розробники даної нейромережі:

- прописувати максимально точний та зрозумілий запит. Уникати абстракцій. Якщо потрібно згенерувати конкретне місце, краще писати його назву замість «відома статуя в Нью-Йорку»;

- не використовувати частки "не", "крім", "без" і схожі. Наприклад, замість «людина не радіє» писати «людина сумує»;

- розбивати складні запити на частини за допомогою ком або подвійних двокрапок «::». Наприклад, так відокремлюють запит об'єкта від стилістики та команд для ШІ.

- запитувати одиночні об'єкти чи вказувати конкретну кількість.

В [30] вказується думка, що для створення картинок слід використовувати архетипи. Особливо, якщо не виходить згенерувати потрібний об'єкт або позу. Наприклад, замість опису «людина з цибулею та стрілами за спиною» краще писати «лучник», замість «людина у халаті зі стетоскопом та документами» – «лікар».

Якщо не виходить і так, то приклади потрібних картинок варто пошукати в Інтернеті, наприклад, Pinterest або на фотобанках. І взяти із цих картинок їх описи. Так нейромережі буде простіше зрозуміти задум, адже саме на таких картинках з описами вони навчалися генерувати зображення.

3.2 Вибір об'єкта

Починати слід з того, що ми хочемо отримати на майбутній картинці. Наприклад, kota (рис.3.1) собаку, лист, літак чи магів. У нейромережі

знайдуться варіанти зображень на будь-який смак, тому що вони навчаються на базі з Інтернету.

Якщо потрібно задати кілька об'єктів, слід розділити їх двома двокрапками "::". Так неймережа не заплутається і намалює те, що потрібно.



Рисунок 3.1 – Реакція неймережі Openjourney на запит “cat”

Унікальність та різноманітність підсумкових картинок вийде у тому випадку, якщо поєднати кілька об'єктів на одному фото. Наприклад, кіт-супергерой (рис.3.2) або людина-біткоїн.



Рисунок 3.2 – Реакція Openjourney на запит “cat superhero in the city”

Крім того, об'єкти можна уособлювати або задавати їм текстуру. Наприклад, створимо Машу та Ведмедя, якби вони були президентами США (рис.3.3). Ще одна можливість – малювати об'єкти так, ніби вони є відомими персонажами.



Рисунок 3.3 – Реакція Openjourney на запит “Masha and the Bear from the cartoon, if they were US presidents”

Наприклад, щоби реалізувати неіснуючий персонаж з анекдоту про помісь Чебурашки і Термінатора – «Чебуратор», введемо запит «Cheburashka from the cartoon, if he were a terminator» (рис.3.4).



Рисунок 3.4 – Реакція Openjourney на запит “ Cheburashka from the cartoon, if he were a terminator ”

Якщо потрібно, щоб об'єктів на картинці було кілька, треба вказати про це у запиті. Наприклад, не кілька папуг, а три папуги (рис.3.5). Але слід враховувати, що чим більше об'єктів потрібно генерувати нейромережі, тим більше страждатиме деталізація кожного об'єкта.

Головне, щоб запити були чіткими та конкретними. Наприклад, на запит «дме вітер» нейромережа малює щось абстрактне (рис.3.6).



Рисунок 3.5 – Реакція Openjourney на запит «Three parrots are sitting on a branch and looking into the sunset»



Рисунок 3.6 – Реакція Openjourney на запит "The wind is blowing"

А на запит «Ураган, вітер дме в полі і гне дерево» відчувається сила природи (рис.3.7).



Рисунок 3.7 – Реакція Openjourney на запит "A hurricane, the wind blows into the field and bends the tree"

Крім реальних об'єктів, нейромережа Openjourney дозволяє генерувати абстракції. Наприклад, майбутнє, час чи кохання (рис.3.8).



Рисунок 3.7 – Реакція Openjourney на запит "Love"

Але отримати якісні зображення на такі теми вдасться рідко – нейромережа плутається і малює не завжди щось виразне. Натомість абстракції краще замінювати об'єктами. Наприклад, не просто кохання, а обійми молодих людей з ніжним заднім фоном (рис.3.8).



Рисунок 3.8 – Реакція Openjourney на запит «A guy and a girl hug on a soft pink background»

Як видно з рис.3.8, у нейромереж на даний час виникають проблеми з руками, особливо пальцями. Їх часто буває більше, або менше, ніж треба. Інколи мережа малює персонажів з більшою кількістю рук або ніг. Приклади подібних ситуацій, що називають «багатопалістю» та «багатоногостю», показані на рис.3.9.



Рисунок 3.9 – Некоректна генерація кінцівок нейромережами

3.3 Додавання деталей

Деталі потрібні, щоб доповнити картинку і зробити її більш виразною. Якщо не прописати жодних деталей, нейромережа додумуватиме їх сама – і іноді вони виходитимуть не зовсім у тему.

Ось які деталі можна додавати: як розташовані об'єкти, що знаходяться в центрі, а що – на задньому плані.

Додавати деталі на зображення можна за допомогою іменників, дієслів та прикметників. Іменники потрібні для того, щоб задати сам об'єкт та його оточення. Наприклад, машина на парковці.

Дієслова допоможуть показати дії. При цьому краще використовувати конкретні дієслова: замість «очікує» – «стоїть і чекає», замість «розчаровується» – «плаче» або «засмучений, тримає руки в обличчя».

А прикметники потрібні, щоб оживити картинку та додати до неї контексту. Наприклад, прикметник «реалістичний» роблять картинку наближеною до реальності. А «старий» може зробити картинку у ретро-форматі.

Зробимо запит для прикладу: «У полі стоїть старий дерев'яний будинок. Поруч стоїть одне велике дерево, одна лавка. На лавці сидить чоловік похилого віку. Захід сонця». Реакція нейромережі Openjourney на запит «Old wooden house in the field :: There is one big tree nearby, one bench :: An elderly man is sitting on a bench :: Sunset» показано на рис. 3.10.



Рисунок 3.10 – Реакція Openjourney на запит «Old wooden house in the field :: There is one big tree nearby, one bench :: An elderly man is sitting on a bench :: Sunset»

Це перша спроба – і дуже вдала. Щоправда, замість лавки людина сидить на паркані, а дерево практично росте з дому. Найвдалішою вийшла перша картинка, і на її прикладі можна запитати неймережу згенерувати ще. Для цього потрібно натиснути кнопку V1 у чаті дискорду. Результат показано на рис.3.11.



Рисунок 3.11 – Реакція Openjourney на уточнення по V1

Треба також не перебільшити з обсягом опису. У Openjourney можна написати до 4000 символів, DALL-E 2 - до 400 символів, у Stable Diffusion - до 75 слів.

3.4 Визначення стилю

У базах нейромереж є сотні зображень із фірмовими стилями будь-яких художників чи жанрів. Тому є можливість згенерувати зображення в стилі Ван Гога, в жанрі нуар або поєднати все – і отримати картину в стилі Марка Шагала у жанрі кіберпанку. Наприклад зробимо запит «Ейфелева вежа в стилі нуар, ніби її намалював Ван Гог». Результат показано на рис.3.12.



Рисунок 3.12 – Реакція Openjourney на запит "The Eiffel Tower in the noir style by Van Gogh"

Крім жанрів змінити стилістику можна за допомогою різного освітлення, формату підсумкової картинки, параметрів камери та об'єктива. Такий етап називається створенням мудборду. Мудборд – це інструмент, який допомагає дизайнеру визначити загальний настрій майбутнього проекту.

Особливості освітлення. Один і той же сюжет можна зробити різним за настроєм та емоціями залежно від колірної схеми. Наприклад, похмурішим і сумнішим, або яскравим і веселим.

На рис.3.13 показана реакція нейромережі Openjourney на запит «Ейфелева вежа в яскравий день із чудовим сонячним освітленням».

На рис.3.14 показана реакція нейромережі Openjourney на запит «Ейфелева вежа в похмурий день, тьмяне освітлення».



Рисунок 3.13 – Реакція Openjourney на запит «Eiffel Tower on a bright day with beautiful sunlight»



Рисунок 3.14 – Реакція Openjourney на запит «Eiffel Tower on a gloomy and overcast day, dim lighting»

Формат підсумкової картинки. Можна попросити неймережу, щоб готове зображення було як картини, арту, фотографії, скульптури тощо.

Різні параметри камери та об'єктива. За допомогою спеціальних команд можна встановити параметри камери або об'єктива. Це вплине на підсумкове зображення. Наприклад, можна отримати реалістичну фотографію, якщо попросити неймережу згенерувати фотографію, зняту на об'єктив 9 мм. Можна додати ефект fish eye (рис.3.14).

Крім цього, можна встановлювати фокусну відстань об'єктива, ISO, експозицію, витримку та інші параметри. Для прикладу можна додати URL-адреси зображень, що сподобалися, перед текстом запити. Неймережа орієнтуватиметься на них під час генерації.



Рисунок 3.15 – Реакція Openjourney на запит «Car :: beach :: the sun is shining :: fisheye effect»

3.5 Огляд найпопулярніших команд для роботи з Openjourney

Команди потрібно додавати в кінці запиту, краще перед цим поставити двокрапку з пробілом (:) – так нейромережа точно зрозуміє, яку команду їй дають.

Популярні команди.

--q <число>. Задає ступінь опрацювання зображень. Тут можна використовувати діапазон від 0,25 до 5. Базове значення стоїть на 1, а на 5 нейромережа пророблятиме кожну картинку хвилин 5.


--s <число>. Показує, як сильно потрібно стилізувати зображення. Значення можна вказувати від 600 до 20 000. Значення за замовчуванням - 2500. Наприклад, якщо задати стилізацію в стилі Ван Гога, то чим більше число, тим сильніше зображення буде схоже на роботи художника.

--chaos <число>. Додає рівень абстракції. Приймає значення від 0 до 100. Чим більше число, тим сильніший рівень абстракції.

:: <число>. Надає вагу якоїсь частини зображення щодо тексту. Значення за замовчуванням – 0,25. Якщо для запиту "email dog" прописати значення email::5 dog:3, то вага у слова email буде більша і при генерації воно буде в пріоритеті. Докладніше про роботу з вагами читайте у документації від Openjourney.

`--iw <число>`. Задає рівень схожості зображень. Наприклад, якщо ви дасте нейромережі приклад картинки і хочете зробити щось схоже – додайте цю команду та бажаний ступінь відповідності. Значення за замовчуванням – 0,25.

`--no`. Відсіває задані слова із запиту. Допомагає прибрати з зображення елементи, які вам не подобаються. Наприклад, команда `-no birds` прибере птахів із зображення.

`--video`. Зберігає відео прогресу генерації. Щоб зберегти відео, потрібно відреагувати конвертом  на повідомлення від нейромережі. Після цього ви отримаєте посилання на відео в особистому листуванні з роботом.

Інші команди.

`/settings` – відкриває поле зі спрощеними налаштуваннями генерації, якості, приватності тощо.

`/prefer option set <ім'я пресета> <параметри>` — дозволяє створити власний шаблон для майбутніх генерацій. Наприклад, `/prefer option set mine --ar 4:3` створить налаштування `--mine` з опцією у співвідношенні сторін 4 на 3. Це зручно, тому що замість вказівки десятків команд можна буде обійтися однією.

`/prefer option list` — показує збережені шаблони. Ви можете зберегти не більше 20 особистих опцій.

За замовчанням картинка генерується у співвідношенні сторін 1:1. Але розмір можна змінити та зробити зображення у різних форматах. Для цього потрібно прописати команди:

`--w` — Ширина зображення. Працює краще, якщо число кратне 64.

`--h` — Висота зображення. Працює краще, якщо число кратне 64.

`--ar` — Бажане співвідношення сторін замість ручного налаштування висоти та ширини за допомогою `--h` та `--w`.

Наприклад, `--ar 16:9`, щоб увімкнути співвідношення сторін 16:9.

Якщо потрібно вказати розмір користувача, слід використати команду `--w X --h Y`.

Примітка: немає зможете налаштувати будь-які співвідношення сторін, які заманеться. Можна вибрати лише стандартні цифри роздільної здатності.

Також можна вказати вихідну роздільну здатність за допомогою ключових слів на кшталт 8K, 4K, photoreal, ultra photoreal, ultra detail, intricate details і т.д.

3.6 Експеримент з фотореалістичними портретами

Для роботи з фотореалістичними портретами можна вказати неймережі на референс фото, що буде «підказкою» для генерації. В нашому експерименті в якості референсу було взято фото завідувача кафедри МІРЕС професора В.М. Карташова з його сторінки на офіційному сайті ХНУРЕ (рис.3.16, а).

Задачею експерименту було змінити вік в меншу і більшу сторону, а також поекспериментувати зі стилем портрету, та обставинами, в яких знаходиться об'єкт на фото. На рис. 3.16 – рис. 3.18 показані результати даного експерименту.



а)

б)

в)

Рисунок 3.16 – Референс (а) і реакція Openjourney на запити: б – «Man in town», в – «Man with beard»



а)

б)

в)

Рисунок 3.17 – Референс (а) і реакція Openjourney на запити: б – «Man at 75 years old», в – «Man at 25 years old»



а)

б)

в)

Рисунок 3.18 – Реакція Openjourney на запит «Stylish man»: а – «Photorealistic», б – «Cyberpunk», в – «Anime»

Отже, можна стверджувати, що поки генеративні нейромержі не можуть зробити два однакових зображення персонажу. В різних генерованих зображеннях персонаж хоч трішки, але буде відрізнятися. Це робить неможливим генерацію картинок, наприклад для коміксів або для анімації.

3.7 Висновки по розділу 3

Генеративно-змагальні мережі були представлені в 2010 році, але найбільший прогрес можна спостерігати з 2022 року.

Спрощено принцип роботи генеративних нейромереж Text-to-Image полягає в таких кроках:

- основою для генерації є випадковий шум;
- застосування процесу зворотної дифузії, тобто обернення процесу щезнення зображення при новому і новому додаванні шуму;
- для правильного формування процесу зворотної дифузії потрібно навчити модель на великій кількості тестових зображень;
- коли модель навчена, вона може сама генерувати зображення за запитом.

Слід зазначити, що генеративні нейромережі не колажують зображення, компілюючи їх з частин вже відомих зображень. Вони кожен раз генерують оригінальне зображення. Двох абсолютно однакових зображень не буде, бо в основі є випадковий шум.

Негативні сторони генеративних нейромереж:

- завжди різні зображення, це робить неможливим генерацію картинок, наприклад для коміксів або для анімації;
- дуже важко, або взагалі неможливо, отримати зображення конкретного об'єкта; можна отримати тільки максимальну схожість, але не точно такий же об'єкт;
- неминуча поява артефактів на кшталт багатопальцевості або багатоногості, доки що ця проблема остаточно не вирішена.

До позитивних сторін віднесемо такі:

- немає проблем з плагіатом, авторські права належать тому, хто робив запит;
- згенерувати можна все, що завгодно, вся справа в часі на підбір вірного запиту;
- доступні стилі всіх художників і фотохудожників світу.

ВИСНОВКИ

Перші нейронні мережі у вигляді комп'ютерного коду були реалізовані ще в 1950-х роках. Тоді вони вирішували прості задачі класифікації, на більш складні завдання не вистачало обчислювальної потужності. Новий поштовх нейромережі отримали в 2010-х роках, коли почали вирішувати задачі аналізу великих об'ємів даних і прогнозування, розпізнавання та синтезу тексту або мови, комп'ютерного зору і багатьох інших задач.

Сьогодні багато програмних продуктів включають спеціальні модулі зі штучним інтелектом, що дозволяють виконувати не тільки обробку звукових, графічних та відеоданих. Вершиною складності є генерація нейромережею власного графічного контенту на основі запиту. Генеративно-змагальні мережі були представлені в 2010 році, але найбільший прогрес можна спостерігати з 2022 року.

Генеративні нейромережі працюють за принципом «вигадали запит, ввели його та отримали відповідь». Але все не так просто: якщо прописати абстрактне питання або перевантажити його деталями, нейромережа заплутається і видасть у відповідь «кашу».

Сучасний спеціаліст в сфері мультимедіа має знати і вміти, як правильно прописувати запити та отримувати коректні відповіді. Нейромережі можна розглядати як повноцінний інструмент створення мультимедійного контенту, поряд з іншими, класичними методами обробки. Як показує практика, сьогодні не існує точних рекомендацій для побудови структури, процесу навчання і критеріїв якості нейронних мереж. Робота з нейромережею лишається дуже суб'єктивним і творчим процесом. Тут допомагає досвід та базові знання про функціонування даного інструменту.

Метою кваліфікаційної роботи є теоретичний аналіз основ побудови, функціонування, навчання та оцінки якості навчання нейронних мереж для обробки і генерації фотореалістичних зображень, а також практичне

дослідження особливостей роботи і результатів, що дають на сьогоднішній день такі нейромережі.

В першому розділі проведено аналіз методів побудови, принципів роботи і навчання нейронних мереж. В основі є нейрон, що додає вхідні сигнали з певними вагами, вихід суматора йде на функцію активації, що перетворює цей сигнал на певне число на виході нейрона. Нейрони об'єднуються певним чином в мережу через синапси – ваги, що показують, яким чином нейрони будуть діяти один на одного.

Нейронів в нейромережі багато, тому вони об'єднуються в шари. Розрізняють вхідний і вихідний шари, де відбувається перетворення вхідних і вихідних даних в сигнали і навпаки, а також приховані шари, де відбувається обробка. Чим більше шарів у нейронній мережі, тим складніше завдання, з якими вона може справлятися. В залежності від поєднання нейронів розрізняють одно-, багат шарові нейронні мережі, мережі прямого поширення, згорткові мережі, мережі зі зворотними зв'язками.

Важлива перевага нейромереж – можливість роботи з будь-якими даними. Головне – процес навчання. Це пошук такого набору вагових коефіцієнтів, у якому вхідний сигнал після проходження мережею перетворюється на потрібний нам вихідний. Багато в процесі навчання залежить від обсягу та представництва навчальної вибірки. Таким чином працюють нейромережі для аналізу і класифікації даних, зокрема, зображень.

В другому розділі розглянута задача генерації зображень нейромережами. Вона є зворотною до класифікації. Якщо для класифікації використовують згортку, в результаті якої на кожному новому етапі розрізнення зображення зменшується вдвічі, то для генерації треба застосовувати операцію зворотної згортки, коли розрізнення зображення буде збільшуватися.

Якщо зовсім спростити процес генерації, то можна сказати так. В основі генерації зображень лежить випадковий шум. Потім застосовується

процес зворотної дифузії, тобто обернення процесу щезнення зображення при новому і новому додаванні шуму.

Щоби навчити процес зворотної дифузії, використовують дві нейронні: одна мережа – генератор, друга – класифікатор (або дискримінатор) зображень. Завдання дискримінатора у тому, щоб відрізнити вихідні зображення генератора від реальних зображень. Задача ж генератора в тому, щоб обдурити дискримінатор, створюючи зображення максимально схожі на зображення в наборі даних. Можна сказати, що генератор і дискримінатор є противниками у процесі. Звідси походить назва: генеративно-змагальна мережа.

Існує вже багато різновидів генеративно-змагальних мереж. Вони усівають недоліки своїх попередників. Наприклад, генератор може обдурювати класифікатор, створюючи одне універсальне зображення, на цьому процес навчання закінчується. Ця ситуація називається колапсом. Або коли треба в одному зображенні згенерувати декілька різних об'єктів, потрібна попередня розмітка, що також призвело до появи нових мереж.

В третьому розділі виконано практичне дослідження особливостей генерації фотореалістичних зображень нейромережами. Дослідження проведено на прикладі сервісу Midjourney. Оскільки він є платним, встановлено його безкоштовний варіант Openjourney, що використовує такий самий змагальний алгоритм. Запит до нейромережі, по суті, можна розцінювати як текстову команду на англійській мові. Причому, точних правил не існує, є тільки загальні рекомендації і власний досвід.

Досліджено вибір об'єкту. Треба точно вказати мережі, що нам потрібно: кіт, собака, дерево, літак. Якщо потрібно задати кілька об'єктів, слід розділити їх двома двокрапками. Унікальність та різноманітність зображень вийде у тому випадку, якщо поєднати кілька характеристик об'єкту. Наприклад, кіт-супергерой.

Крім того, об'єкти можна уособлювати або задавати їм текстуру. Наприклад, створено Машу та Ведмедя, якби вони були президентами США.

Щоби реалізувати неіснуючий персонаж з анекдоту про помісь Чебурашки і Термінатора – «Чебуратор», введено запит «Cheburashka from the Soviet cartoon, if he were a terminator».

Якщо потрібно, щоб об'єктів на картинці було кілька, треба вказати про це у запиті. Наприклад, не кілька папуг, а три папуги. Але чим більше об'єктів потрібно генерувати нейромережі, тим більше страждатиме деталізація кожного об'єкта.

Запити мають бути чіткими та конкретними. Наприклад, на запит «дме вітер» нейромережа малювала щось абстрактне. А на запит «Ураган, вітер дме в полі і гне дерево» відчувається сила природи. Звісно, нейромережа Midjourney дозволяє генерувати абстракції. Наприклад, майбутнє, час чи кохання. Але результати можуть бути зовсім непередбачуваними.

Абстракції краще замінювати об'єктами. Наприклад, не просто кохання, а обійми молодих людей з ніжним заднім фоном. Як видно з отриманих зображень, у нейромереж на даний час виникають проблеми з руками, особливо пальцями. Їх часто буває більше, або менше, ніж треба. Іноколи мережа малює персонажів з більшою кількістю рук або ніг. Поки що ця проблема остаточно не вирішена.

Щоб доповнити картинку і зробити її більш виразною, потрібні деталі. Якщо їх не прописати, нейромережа додумає їх сама – і іноді вони будуть не в тему. Додавати деталі на зображення можна за допомогою іменників, дієслів та прикметників. Іменники потрібні для того, щоб задати сам об'єкт та його оточення. Наприклад, машина на парковці. Дієслова допоможуть показати дії. При цьому краще використовувати конкретні дієслова: замість «очікує» – «стоїть і чекає», замість «розчаровується» – «плаче» або «засмучений, тримає руки в обличчя».

А прикметники потрібні, щоб оживити картинку та додати до неї контексту. Наприклад, прикметник «реалістичний» роблять картинку наближеною до реальності. А «старий» може зробити картинку у ретро-форматі.

Для експерименту зроблено запит: «У полі стоїть старий дерев'яний будинок. Поруч стоїть одне велике дерево, одна лавка. На лавці сидить чоловік похилого віку. Захід сонця». Перша спроба була дуже вдала. Щоправда, замість лавки людина сидить на паркані, а дерево практично росте з дому. Найвдалішою вийшла перша картинка, і на її прикладі був згенерований новий, більш вдалий варіант.

Дуже важливим для створення фотоконтенту є можливість налаштування освітлення. Отримані зображення на запити «Ейфелева вежа в яскравий день із чудовим сонячним освітленням» і «Ейфелева вежа в похмурий день, тьмяне освітлення». Крім цього, можна встановлювати фокусну відстань об'єктива, ISO, експозицію, витримку та інші параметри камери.

Для роботи з фотореалістичними портретами можна вказати неймережі на референс фото, що буде «підказкою» для генерації. В нашому експерименті в якості референсу було взято фото завідувача кафедри МІРЕС професора В.М. Карташова з його сторінки на офіційному сайті ХНУРЕ. Задачею експерименту було змінити вік в меншу і більшу сторону, а також поекспериментувати зі стилем портрету, та обставинами, в яких знаходиться об'єкт на фото.

По результатам, можна стверджувати, що поки генеративні неймережі не можуть зробити два однакових зображення персонажу. В різних генерованих зображеннях персонаж хоч трішки, але буде відрізнятися. Це робить неможливим генерацію картинок, наприклад для коміксів або для анімації.

Проте, кожне зображення є унікальним, що повністю усуває проблему авторського права. Авторські права належать тому, хто робив запит.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Субботін С.О. Нейронні мережі : теорія та практика: навч. посіб. – Житомир: Вид. О. О. Євенок, 2020. – 184 с.
2. Руденко, О. Г. Штучні нейронні мережі / О. Г. Руденко, Є. В. Бодянський. – Харків : Компанія СМІТ, 2006. – 404 с.
3. С. А. Субботин, Ан. А. Олейник, Е. А. Гофман, С. А. Зайцев, Ал. А. Олейник. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов : монография / под ред. С. А. Субботина. – Харьков : Компания СМІТ, 2012. – 318 с.
4. Schmidhuber, J. Deep Learning in Neural Networks: An Overview // Neural Networks. – 2015. – Vol. 61. – P. 85–117.
5. Bengio, Y. Deep Learning / Y. Bengio, Y. LeCun, G. Hinton // Nature. – 2015. – Vol. 521. – P. 436–444.
6. Gers, F. Learning precise timing with LSTM recurrent networks / F. Gers, N. Schraudolph, J. Schmidhuber // Journal of Machine Learning Research. – 2002. – Vol. 3. – P. 115-143.
7. Generative Adversarial Networks [Electronic resource] / I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, Sh. Ozair, A. Courville, Aaron, J. Bengio. – Access mode: <https://arxiv.org/pdf/1406.2661.pdf>.
8. Demuth, H. Neural Network Toolbox for use with Matlab: user's guide / H. Demuth, M. Beale. – Natick: Mathworks Inc, 1997. – 700 p.
9. Sumathi, S. Computational intelligence paradigms: theory and applications using Matlab / S. Sumathi, S. Paneerselvam. – Boca Raton: CRC Press, 2010. – 851 p.
10. Субботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей : монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2009. – 375 с.

11. Chen X., Xie L., Wu J., Tian Q. Cyclic CNN: Image Classification with Multi-Scale and MultiLocation Contexts. *IEEE Internet of Things Journal*, 2020.
12. Dais D., Bal I., Smyrou E., Sarhosis V. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Automation in Construction*, 2021.
13. Yang Q. Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, 2018. – pp. 1348–1357.
14. Zubkov O.V., Sheiko S.O., Oleynikov V.M., Kartashov V.M., Babkin S.I. Investigation of Efficiency of Detection and Recognition of Drone Images from Video Stream of Stationary Video Camera / *Telecommunications and Radio Engineering*, 2021, 80(3), pp. 23–37.
15. Koryttsev, S. Sheiko, V. Kartashov, O. Zubkov, V. Oleynikov, I. Selieznov, M. Anohin. Practical Aspects of Range Determination and Tracking of Small Drones by Their Video Observation // 2020 International Scientific-Practical Conference. Problems of Infocommunications. Science and Technology. Kharkiv, Ukraine. October 6-9, 2020. – 5 p.
16. V. Kartashov, V. Oleynikov, O. Zubkov, S. Sheiko. Optical detection of unmanned air vehicles on a video stream in a real-time // *The Fourth International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo'2019)*, 9–13 September 2019, Odessa, Ukraine, 4 p.
17. В.М. Карташов, И.В. Корытцев, С.А. Шейко, В.Н. Олейников, О.В. Зубков, С.И. Бабкин. Оптико-электронные методы обнаружения воздушных объектов и измерения их координат // *Радиотехника: Всеукр. межвед. науч.-техн. сб.* – 2020. – Вып. 202. – С. 153 – 159.
18. О.В. Зубков, С.А. Шейко, В.Н. Олейников, В.М. Карташов, И.В. Корытцев, С.И. Бабкин. Исследование эффективности детектирования и распознавания изображений дронов по видеопотоку // *Радиотехника: Всеукр. межвед. науч.-техн. сб.* – 2020. – Вып. 202. – С. 136 – 146.

19. И.В. Корытцев, С.А. Шейко, В.М. Карташов, О.В. Зубков, В.Н. Олейников, С.И. Бабкин, И.С. Селезнев. Обработка сигналов при пеленгации и определении дальности до малоразмерных БПЛА в оптическом и инфракрасном диапазонах // Радиотехника: Всеукр. межвед. науч.-техн. сб. – 2020. – Вып. 202. – С. 125 – 134.
20. Zubkov O., Sheiko S., Oleynikov V., Kartashov V. Investigation of the neural networks effectiveness in recognizing moving drones. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2021, 2, pp. 119–122.
21. Sytnik O., Kartashov V. Technical vision model of the visual systems for industry application // Examining Optoelectronics in Machine Vision and Applications in Industry 4.0, 2021, pp. 288–310.
22. Довгаль К.С. Особливості роботи та використання програм тривимірного простору // 24-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 3. – Харків: ХНУРЕ. 2021. – с. 67 – 68.
23. Тимченко Г.О. Оцінка дальностей виявлення, розпізнавання та ідентифікації БПЛА в оптико-електронному каналі // 25-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 3. – Харків: ХНУРЕ. 2021. – с. 45 – 46.
24. Ключенков Р.О. Дослідження адаптивних методів подавлення шуму в відеозображеннях // 24-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 3. – Харків: ХНУРЕ. 2020. – с. 111 – 112.
25. Кац Д.В. Ефективність методів віднімання фону в задачах виявлення малих БПЛА // 24-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 3. – Харків: ХНУРЕ. 2020. – с. 101 – 102.
26. T. Park, M. -Y. Liu, T. -C. Wang and J. -Y. Zhu, "Semantic Image Synthesis With Spatially-Adaptive Normalization," 2019 IEEE/CVF Conference

on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 2332-2341.

27. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

28. Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein Generative Adversarial Networks / *Proceedings of the 34th International Conference on Machine Learning*, PMLR 70, 2017. – p. 214-223,

29. Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. – p. 2813-2821.

30. Midjourney Documentation. [Электронный ресурс]. URL: <https://docs.midjourney.com/> (дата звернення 11.12.2023).