

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РОЗРОБКА МОДЕЛІ І МЕТОДУ ГЕНЕРУВАННЯ
ЗОБРАЖЕНЬ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ
(тема)

Виконав:
студент 2 курсу, групи ІНФМ-18-2

Вовк А. С.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Путятін Є. П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Путятін Є.П.
(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« 21 » жовтня 2019 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Вовк Андрію Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розробка моделі і методу генерування зображень на основі нейронних мереж

затверджена наказом по університету від « 21 » жовтня 2019 року № 1056Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 25 листопада 2019 р.

3. Вихідні дані до роботи: вибірка зображень для навчання нейронної мережі, програмна реалізація нейронної мережі, Wasserstein GAN, Softmax GAN, OpenAI Gym, Python, Linux (CentOS).

4. Перелік питань, що потрібно опрацювати в роботі: _____

1. Основні відомості нейронних мереж

2. Фільтрація зображень

3. Проблеми тренування нейронних мереж

4. Генерація зображень

5. Варіанти застосування

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів): Схема роботи згорткової нейронної мережі, порівняння результату алгоритму SPADE з результатами алгоритму ріх2ріхHD, порівняння високорівневого алгоритму генератора SPADE з генератором ріх2ріхHD, схема роботи автокодерів.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	21.10.2019	
2	Аналіз завдання, підбір літератури	05.11.19-10.11.19	
3	Аналіз літератури з досліджуваної проблеми	11.11.19-15.11.19	
4	Аналіз технічних засобів	16.11.19-18.11.19	
5	Розробка методу	19.11.19-22.11.19	
6	Програмна реалізація	22.11.19-24.11.19	
7	Оформлення пояснювальної записки	25.11.19-01.12.19	
8	Перевірка на плагіат	22.11.19	
9	Рецензування	23.11.19	
10	Підготовка презентації та доповіді	23.11.19	
11	Занесення роботи в електронний архів	24.11.19	
12	Попередній захист атестаційної роботи	25.11.19	

Дата видачі завдання 21 жовтня 2019 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Путятін Є. П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до атестаційної роботи: 77 с., 2 табл., 30 рис., 1 дод., 44 джерела.

ГЕНЕРАТИВНО-ЗМАГАЛЬНІ НЕЙРОМЕРЕЖІ, НЕЙРОМЕРЕЖІ, IMAGE-TO-IMAGE, PIX2PIXHD, SPADE, 2D ЗОБРАЖЕННЯ, ГРАФІЧНИ ОБРАЗИ.

Метою роботи є розробка методів, що базуються на використанні нейромереж, які дозволяють генерувати нові зображення на основі вивчених зображень. GAN досягають цього рівня реалізму, поєднуючи генератор, який вчиться виробляти цільовий вихід, з дискримінатором, який вчиться відрізняти справжні дані від виходу генератора. Потенціал GAN величезний, оскільки вони імітують будь-який розподіл даних. GAN навчають створювати структури, страхотливо схожі на сутності з нашого світу в області зображень, музики, мови, прози.

Об'єктом дослідження є процедури генерації зображень.

Розробка, проектування і тестування нейронної мережі для полегшення процедури розпізнавання графічних образів. Дослідження методів розпізнавання графічних образів з використанням нейронних мереж. Розробка комп'ютерної системи генерації графічних образів.

У результаті роботи здійснена програмна реалізація системи для процедури розпізнавання графічних образів.

GENERATIVE ADVERSARIAL NETWORKS, NEURAL NETWORK, IMAGE-TO-IMAGE, PIX2PIXHD, SPADE, 2D IMAGE, GRAPHIC IMAGES.

The goal is to develop neural network-based methods that allow the generation of new images based on the studied images. GANs achieve this level of realism by pairing a generator, which learns to produce the target output, with a discriminator, which learns to distinguish true data from the output of the generator. The potential of GAN is huge, then they mimic any data extension. GAN footwear creates structures using similar entities from our world in the field of images, music, things, prose.

The object of study is a sequence of different images of objects.

Design, design and testing of a neural network to facilitate graphic recognition. Investigation of methods of graphic pattern recognition using neural networks. Development of computer graphic recognition system.

As a result, the software implementation of the system for the procedure for recognizing graphic images was carried out.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Основні відомості нейромереж.....	10
1.1 Введення в машинне навчання.....	10
1.2 Класичні методи машинного навчання, розпізнавання образів і комп'ютерного зору.....	11
1.3 Фільтрація.....	12
1.3.1 Бінаризація по порогу, вибір області гістограми.....	12
1.3.2 Перетворення Фур'є.....	12
1.3.3 Фільтри частот.....	13
1.3.4 Вейвлет-перетворення.....	13
1.3.5 Обчислення показників кореляції.....	13
1.3.6 Фільтри функцій.....	14
1.3.7 Фільтри контурів.....	14
1.3.8 Інші фільтри.....	15
1.4 Визначення нейромереж.....	15
1.5 Класифікація нейронних мереж.....	17
1.5.1 Багатошаровий перцептрон.....	19
1.5.2 Рекурентний перцептрон.....	21
1.5.3 Згорткові нейронні мережі.....	22
1.5.4 Мережа радіально-базисних функцій.....	24
1.5.5 Повно зв'язні нейронні мережі.....	25
1.5.6 Генеративно-змагальні мережі.....	26
1.5.7 Мережа Хопфілда.....	28
1.5.8 LSTM И GRU.....	29
1.5.9 Мережі Джордана і Елма.....	30
1.5.10 Само організаційна карта Кохонена.....	31
1.5.11 Мережа Хеммінга.....	33

	6
1.6 Постановка задачі дослідження.....	34
2 Структура та методи генеративно-змагальних нейромереж.....	42
2.1 Структура генеративно-змагальних нейронних мереж.....	42
2.1.1 Генератор.....	42
2.1.2 Дискримінатор.....	43
2.1.3 Як працюють генеративно-змагальна нейронна мережа.....	44
2.1.4 Автокодери і VAE.....	46
2.1.5 Поради з навчання GAN.....	47
2.2 Проблеми тренування.....	48
2.3 Як і де застосовуються GAN і генеративні моделі.....	49
2.4 Генеративно-змагальна нейронна мережа Spade.....	54
2.5 Висновок і подальші дослідження.....	56
3 Комп'ютерна модель генерації зображень.....	57
3.1 Прогресивне навчання генеративно-змагальної мережі.....	57
3.2 Підвищення варіативності шляхом стандартного відхилення.....	59
3.3 Важливість індивідуальних внесків у умовах статистичної безпечності.....	61
3.4 Швидкість конвергенції та навчання.....	66
Висновки.....	68
Перелік джерел посилання.....	70
Додаток а.....	75

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СТЗ – система технічного зору

GAN – Generative Adversarial Networks

НМ – нейронна мережа

Датасет – набір даних, вибірка

Tensorflow – назва бібліотеки машинного навчання

РНМ – рекурентна нейронна мережа

ШНМ – штучна нейронна мережа

БД – база даних

ОС – операційна система

ІТ - інформаційні технології

ВСТУП

Generative Adversarial Networks (GAN) користуються все більшою популярністю. Багато хто говорить про них, хтось навіть вже використовує. Це алгоритм машинного навчання, що входить в сімейство породжують моделей і побудований на комбінації з двох нейронних мереж, одна з яких генерує зразки, інша намагається відрізнити справжні зразки від згенерованих.

GAN складається з одного дискримінатора та одного генератора. Дискримінатор намагається відрізнити реальні зразки від створених зразків, тоді як генератор підробляє реальні зразки, використовуючи інформацію від дискримінатора. GAN унікальна з багатьох інших генеративних моделей. Замість явної вибірки з розподілу ймовірностей GAN використовує глибоку нейронну мережу як прямий генератор, який генерує зразки з випадкових шумів.

Стратегічна підготовка GAN полягає у визначенні гри між двома конкуруючими мережами. Мережа генератора відображає джерело шуму до вхідного простору. Мережа дискримінатора отримує або згенерований зразок, або справжній зразок даних, і повинен розрізняти два. Генератор навчений дурити дискримінатора.

Принцип змагальності в мережі GAN нерідко описується через метафори. Наприклад, генеративна мережа уподібнюється фальшивомонетнику або підроблювач картин, а дискримінатор - експерту який прагне розпізнати підробку.

Інший приклад - образ двох боксерів, один з яких навчався у майстра, а другий змушений наслідувати учневі. У популярному додатку генерації людських обличчя в якості справжніх даних виступають реальні фотографії, а генеративна мережа намагається створити штучні особи, варіюючи комбінації таких латентних параметрів, як колір волосся, пропорції обличчя, розріз очей, форма носа, розмір вух, наявність бороди і вусів і т . д.

GAN можуть створювати зразки, які приваблюють візуально, але часто важко піддаються навчанню, і значна частина останніх робіт з цього питання була присвячена пошуку способів стабілізації навчання. Незважаючи на це, стабільно стабільне навчання GAN залишається відкритою проблемою. Зокрема, надається аналіз властивостей конвергенції значущої функції, оптимізованої GAN. Запропонована ними альтернатива, названа Wasserstein GAN (WGAN), використовує відстань Wasserstein для отримання функції значення, яка має кращі теоретичні властивості, ніж оригінал.

Генеративні моделі застосовуються для створення контенту і даних. Наприклад, створення картинок для інтернет-магазину, аватарів для ігор, відеокліпів, згенерованих автоматично, виходячи з музичного біта твору, або навіть віртуальних провідних для ТВ-програм. Завдяки роботі GAN і генеративних моделей виникає синтез даних, на яких потім будуть навчатися інші системи.

У цій атестаційній роботі буде розглянуто методи генерування зображення на основі нейронних мереж. Основна ідея - автоматизація виробництва контенту, який раніше вимагав зусиль художників, архітекторів, дизайнерів, мультиплікаторів.

1 ОСНОВНІ ВІДОМОСТІ НЕЙРОМЕРЕЖ

1.1 Введення в машинне навчання

Машинне навчання - це великий підрозділ теорії штучного інтелекту, суворо математична дисципліна, яка використовує розділи математичної статистики, чисельних методів оптимізації, теорії ймовірностей [7].

Існує два типи машинного навчання: навчання по прецедентах (індуктивне навчання), засноване на емпіричних даних, і дедуктивний навчання, що припускає формалізацію знань і формування бази знань. Дедуктивне навчання прийнято відносити до області експертних систем, тому в теорії і практиці машинного навчання фактично розглядається навчання по прецедентах.

Розділ машинного навчання виник в результаті поділу науки про нейронних мережах в рамках науки про штучний інтелект на методи навчання мереж і види топології архітектури мереж, увібравши в себе деякі інші області, такі як методи математичної статистики і теорію дискретного аналізу [9].

Специфіка розглянутих способів навчання в рамках дисципліни обумовлена:

- навчання з учителем - для кожного прецеденту існує пара «ситуація, рішення»;
- навчання без вчителя - система групує об'єкти в кластери і знижує розмірність вхідної інформації, використовуючи дані про попарному схожості;
- навчання з підкріпленням - для кожного прецеденту існує пара «ситуація, реакція середовища »(навчання з підкріпленням можна вважати окремим випадком навчання з учителем, так і окремим випадком навчання без вчителя).

Існують також інші, менш поширені способи навчання, наприклад, активне навчання (той, якого навчають алгоритм має можливість призначити

наступну досліджувану ситуацію), часткове залучення вчителя, трансдуктивне навчання, багатозадачність і різноманітне навчання, але відмінності в цих способах несуттєві в рамках даної проблеми.

Всі перераховані способи можна використовувати для класифікації традиційних методів машинного навчання і для класифікації алгоритмів навчання нейронних мереж, що реалізують будь-який з них.

Сучасне машинне навчання стикається з гострою проблемою універсальності, оскільки практично не існує однорідного простору алгоритмів і методу загального вирішення проблеми індукції [10].

1.2 Класичні методи машинного навчання, розпізнавання образів і комп'ютерного зору

Тут і далі теорія комп'ютерного зору, теорія розпізнавання образів, теорія машинного навчання і обробка зображень будуть розглядатися в сенсі розпізнавання образів в комп'ютерному зорі. В іншому випадку буде зазначений інший сенс. З метою стислості може згадуватися тільки одна з цих дисциплін в залежності від контексту.

Класичні методи комп'ютерного зору, розпізнавання образів і машинного навчання можна умовно розділити на три групи:

- методи фільтрації;
- методи аналізу;
- методи навчання.

Основні класичні методи реалізовані в бібліотеці OpenCV, написаної на C / C ++, і займає домінуюче становище при вирішенні задач комп'ютерного зору з використанням класичних підходів.

1.3 Фільтрація

У завданнях комп'ютерного зору фільтрація найчастіше використовується для попередньої обробки зображення перед аналізом його внутрішніх морфологічних ознак, але зустрічаються і завдання, в яких достатнім і бажаним буде використання тільки фільтрації як такої (наприклад, в задачах машинного зору).

1.3.1 Бінаризація по порогу, вибір області гистограми

Для RGB зображення і зображення в градаціях сірого порогом є значення кольору. Вибір порога, за яким відбувається бінаризація, багато в чому визначає процес бінаризація. Зазвичай бінаризація здійснюється за допомогою адаптивного алгоритму, що вибирає поріг. Таким алгоритмом може бути вибір математичного очікування, моди або піків гистограми. При роботі з гистограмою бінаризація ефективна для сегментації квітів.

1.3.2 Перетворення Фур'є

Перетворення Фур'є майже не використовується при обробці зображень в чистому вигляді, оскільки для аналізу зображень одновимірного перетворення зазвичай не вистачає і виникає необхідність використання куди більш ресурсоємних двовимірного перетворення. Цей метод застосовується тільки в разі якщо необхідний аналіз спектра, оскільки використання згортки цікавить області з уже готовим фільтром виявляється швидше і простіше в реалізації. Тим не менше, одномірне перетворення Фур'є застосовується при компресії зображень [13].

1.3.3 Фільтри частот

Найпростіший приклад фільтра низьких частот - фільтр Гаусса, фільтра високих частот - фільтр Габора. Для кожної точки зображення вибирається вікно, в рамках якого виконується твір вихідних даних з фільтром того ж розміру (згортка). Такий підхід отримав досить широке поширення в практиці, дозволяючи виділяти на зображенні цікавиль інформацію, редуцірую зайву інформацію. Зокрема, підхід поширений як одна з реалізацій швидкого шумозаглушення в багатьох областях науки і техніки [13].

1.3.4 Вейвлет-перетворення

Для згортки з сигналом (областю зображення) використовується якась характеристична функція, звана вейвлетом, що визначається як вейвлет-перетворення. Вейвлети - це сімейства функцій, локальних за часом і по частоті, в яких всі функції виходять з однієї з її допомогою зрушень і розтягувань по осі часу. Існує набір класичних функцій, використовуваних в вейвлет-аналізі. До них відносяться вейвлет Хаара, вейвлет Морлі, вейвлет MNat, вейвлет Добеши. На практиці вейвлет-аналізом називається пошук довільного патерну на зображенні за допомогою згортки зображення з моделлю цього патерну. Класичні вейвлети використовуються для стиснення або класифікації зображень [13].

1.3.5 Обчислення показників кореляції

Обчислення кореляції, що лежить в основі вейвлет-перетворення, саме по собі є незамінним інструментом в системах комп'ютерного зору і часто використовується в своєму природному вигляді, наприклад, для знаходження

зрушень або оптичних потоків (Кореляція відео потоку). На основі ізницевого корелятора реалізується найпростіший детектор зсуву [13].

1.3.6 Фільтри функцій

У цьому підході використовуються математичні фільтри, що дозволяють виявляти прості математичні функції на зображенні, для чого формується акумулює зображення (накопичувальний простір), в якому для кожної точки вихідного зображення будується безліч породжують її функцій.

Класичним прикладом є узагальнена перетворення Хафа, що застосовується до бінаризованими зображень, що дозволяє знаходити на зображенні будь-які параметризуємих функції. Модифіковане перетворення Хафа дозволяє шукати будь-які фігури, але в обробці зображень його використання пов'язане з недостатньою стабільністю: висока чутливість до якості бінаризація і низька швидкість роботи змушують шукати цього методу більш ефективні альтернативи. Аналогом перетворення Хафа для прямих є перетворення Радону, що застосовується до небінаризованим зображень і обчислюється через БПФ, що дає значний вигравш в продуктивності на великих масивах даних [11-13].

1.3.7 Фільтри контурів

Фільтри контурів дуже корисні при вирішенні задач комп'ютерного зору в разі якщо необхідна реалізація обробки складних об'єктів на зображенні. Фільтрацію контурів реалізує ряд алгоритмів, таких як: оператор Кенні, який є найбільш часто використовуваним алгоритмом пошуку контурів, оператор Собеля, оператор Лапласа, оператор Прюїтт, оператор Робертса. Алгоритм Кенні реалізує поняття придушення немаксимумов, що означає, що пікселями

кордонів оголошуються пікселі, в яких досягається локальний максимум градієнта в напрямку вектору градієнта. В комп'ютерному зорі і обробці зображень при пошуку меж об'єктів крім особливих окремих випадків важко знайти детектор, який би працював істотно краще, ніж детектор Кенні.

1.3.8 Інші фільтри

Модифікації вищевказаних фільтрів дозволяють вирішити переважна кількість завдань фільтрації на рівні класичних методів, але крім них існують десятки більш спеціалізованих алгоритмів фільтрації, на практиці використовуються значно рідше.

Найбільш цікавими є ітераційні фільтри, такі як активна модель зовнішнього вигляду, ріджлет перетворення, курвлет-перетворення, що виникли на основі класичних вейвлет-функцій і аналізу в поле перетворення Радону, бімлет-перетворення, що реалізує частково завдання фільтрації, частково – завдання аналізу, що дозволяє знаходити контури об'єктів. Ці фільтри володіють рідкісною специфікою і в силу своєї вузької спрямованості не розглядатимуться в рамках даної роботи докладніше.

1.4 Визначення нейромереж

Нейронна мережа - це послідовність нейронів, з'єднаних між собою синапсами. Структура нейронної мережі прийшла в світ програмування прямо з біології. Завдяки такій структурі, машина отримує можливість аналізувати і навіть запам'ятовувати різну інформацію. Нейронні мережі також здатні не тільки аналізувати вхідну інформацію, а й відтворювати її зі своєї пам'яті. Іншими словами, нейронна мережа це машинна інтерпретація мозку людини,

в якому знаходяться мільйони нейронів передають інформацію у вигляді електричних імпульсів.

Найважливішою властивістю нейронних мереж є їх здатність навчатися на основі даних навколишнього середовища і в результаті навчання підвищувати свою продуктивність. Підвищення продуктивності відбувається з часом відповідно до певних правил. Навчання нейронної мережі відбувається за допомогою інтерактивного процесу коригування синоптичних ваг і порогів. В ідеальному випадку нейронна мережа отримує знання про навколишнє середовище на кожній ітерації процесу навчання.

З поняттям навчання асоціюється досить багато видів діяльності, тому складно надати цьому процесу однозначне визначення. Більш того, процес навчання залежить від точки зору на нього. Саме це робить практично неможливим появу будь-якого точного визначення цього поняття. Наприклад, процес навчання з точки зору психолога в корені відрізняється від навчання з точки зору шкільного вчителя.

Нейронні мережі використовуються для вирішення складних завдань, які вимагають аналітичних обчислень подібних тим, що робить людський мозок.

Найпоширенішими застосуваннями нейронних мереж є:

- класифікація;
- передбачення;
- розпізнавання.

Класифікація - розподіл даних по параметрах. Наприклад, на вхід дається набір людей і потрібно вирішити, кому з них давати кредит, а кому ні. Цю роботу може зробити нейронна мережа, аналізуючи таку інформацію як: вік, платоспроможність, кредитна історія. Передбачення - можливість прогнозувати наступний крок. Наприклад, зростання або падіння акцій, ґрунтуючись на ситуації на фондовому ринку. Розпізнавання - в даний час, саме широке застосування нейронних мереж. Використовується в Google, коли

ви шукаєте фото або в камерах телефонів, коли воно визначає положення вашого обличчя і виділяє його і багато іншого [1-2].

Штучна нейронна мережа є концептуальною моделлю біологічної нейронної мережі і складається з пов'язаних різним чином шарів штучних нейронів, які організують загальну активну структуру і функціонально впливають на роботу один до іншого. У більшості архітектур ІНС активність нейрона визначається перетворенням зовнішнього сумарного впливу інших нейронів на даний нейрон.

З моменту свого зародження технології штучних нейронних мереж розвивалися досить відокремленим від класичних методів шляхом, нерідко докорінно змінюючи уявлення про предмет в сукупній проблематики теорій машинного навчання і розпізнавання образів, роблячи значний вплив на теоретичний, термінологічний і методологічний апарати цих дисциплін. З цього часу в науковому співтоваристві сталося кілька спадів і підйомів інтересу до цього напрямку, але, завдяки деяким проривів в теорії ІНС, широке практичне застосування непромережені технології отримали порівняно недавно.

1.5 Класифікація нейронних мереж

Штучні нейронні мережі надзвичайно різноманітні за конфігураціями. Незважаючи на це, мережеві парадигми мають багато спільного.

Нейронні мережі розрізняють за топологічними типами відповідно до структури зв'язків між нейронами мережі, а також за типом використаних формальних нейронів. ШНМ можуть розглядатися як спрямований граф зі зваженими зв'язками, у якому штучні нейрони є вузлами. По архітектурі зв'язків ШНМ можуть бути згруповані в два класи мережі прямого поширення, у яких графи не мають петель, і рекурентні мережі, або мережі зі зворотними зв'язками.

За організацією навчання поділяють навчання нейронних мереж з вчителем (supervised neural networks) та без вчителя (nonsupervised).

За методом навчання поділяють навчання за входами і за виходами. При навчанні за входами навчальний приклад є тільки вектором вхідних сигналів, а при навчанні за виходами до нього входить і вектор вихідних сигналів, який відповідає вхідному вектору.

За методом подання прикладів розрізняють подання одиночних прикладів і "сторінки" прикладів. У першому випадку зміна стану нейронної мережі (навчання) відбувається після подання кожного прикладу. У другому - після подання "сторінки" (множини) прикладів на підставі аналізу одразу усіх їх.

За особливостями моделі нейрону розрізняють нейрони з різними нелінійними функціями.

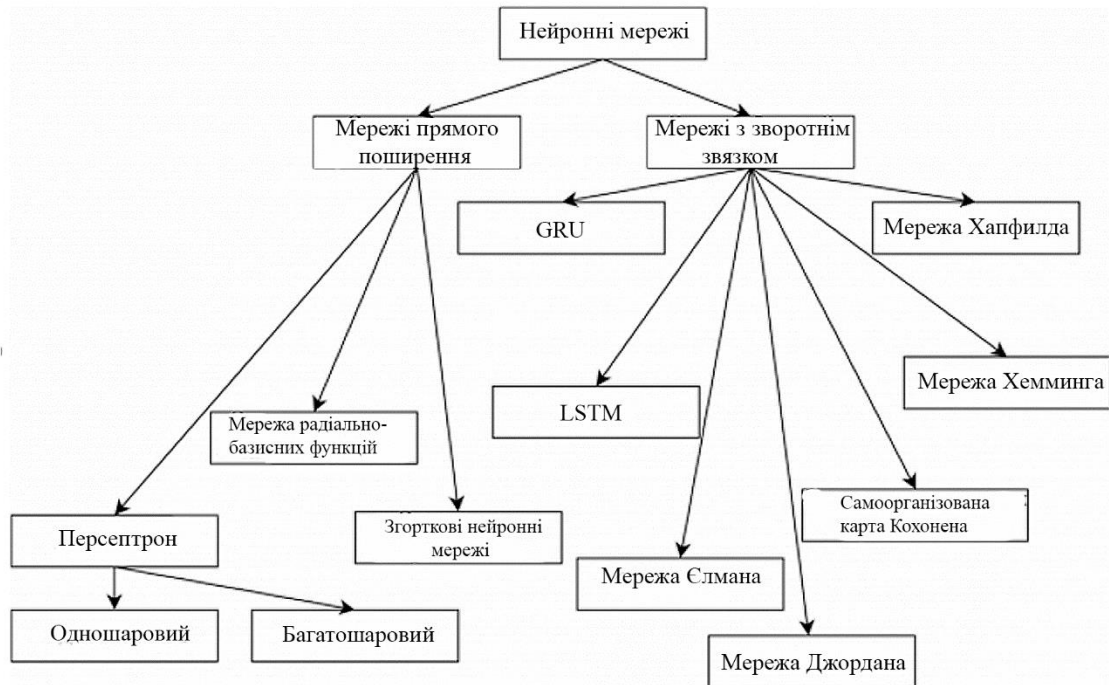


Рисунок 1.1 - Види основних нейронних мереж

1.5.1 Багатошаровий перцептрон

Найвідоміша та дуже стара архітектура, в основі якій йдуть підряд кілька шарів нейронів - вхідний, один або кілька прихованих шарів, і вихідний шар. Нейрони кожного шару з'єднуються з нейронами попереднього і наступного шарів за принципом "кожний з кожним". Кількість нейронів у шарах може бути довільною. Звичайно у всіх схованих шарах однакова кількість нейронів. Майже завжди навчається методом зворотного поширення помилки - що автоматично означає, що ми повинні надати для навчання набір пар «вхідний вектор - правильний вихід». Існує плутанина з підрахунком кількості шарів у мережі. Вхідний шар не виконує ніяких обчислень, а лише розподіляє вхідні сигнали, тому іноді його вважають, іноді — немає.

Тоді вхідний вектор відправиться на вхід мережі, послідовно будуть розраховані стану всіх проміжних нейронів, і на виході утворюється вихідний вектор, який ми і порівняємо з правильним. Розбіжність дасть нам помилку, яку можна поширити назад зі зв'язків мережі, обчислити внесок в підсумкову помилку кожного нейрона, і скорегувати його ваги, щоб її виправити. Повторивши цю процедуру багато тисяч разів, можливо вийде навчити мережу.

Кожен шар розраховує нелінійне перетворення від лінійної комбінації сигналів попереднього шаруючи. Звідси видно, що лінійна функція активації може застосовується тільки для тих моделей мереж, де не потрібно послідовне з'єднання шарів нейронів один за одним.

Для багатошарових мереж функція активації повинна бути нелінійною, інакше можна побудувати еквівалентну одношарову мережу, і багатошаровість виявляється непотрібною. Якщо застосована лінійна функція активації, де кожен шар буде давати на виході лінійну комбінацію входів.

Слідуючий шар дасть лінійну комбінацію виходів попереднього, а це еквівалентно однієї лінійної комбінації з іншими коефіцієнтами, і може бути реалізоване у виді одного шару нейронів.

Багатошарова мережа може формувати на виході довільну багатомірну функцію при відповідному виборі кількості шарів, діапазону зміни сигналів і параметрів нейронів.

Мережа такого типу зазвичай дуже добре справляється з завданнями, де:

- відповідь дійсно залежить тільки від того, що ми даємо на вхід до мережі, і ніяк не залежить від історії входів (тобто це не динамічний процес, або, принаймні, ми дали на вхід вичерпну інформацію про цей процес у формі, яка більш підходить для обробки мережею);
- відповідь не залежить або слабо залежить від високих ступенів і / або творів параметрів – але функції цього типу мережу будувати майже не вміє;
- в наявності є досить багато прикладів (бажано мати не менше сотні прикладів на кожен зв'язок мережі), або у вас є великий досвід боротьби з ефектом спеціалізації.

Це пов'язано з тим, що маючи багато коефіцієнтів, мережа може банально запам'ятати багато конкретних прикладів, і видавати на них відмінний результат - але її прогнози не матимуть нічого спільного з реальністю в разі, якщо дати на вхід приклади не з навчальної вибірки.

Сильні сторони - вивчена з усіх боків, добре працює на своїх завданнях, якщо на деякій задачі не працює - то це привід стверджувати, що завдання складніше, ніж здавалося.

Слабкі сторони - невміння працювати з динамічними процесами, необхідність великої навчальної вибірки.

Перспективи - ніяких істотних. Більшість серйозних завдань, які все ще потребують вирішення, не входять до класу задач, що вирішуються багатошаровим перцептроном с методом зворотного поширення помилки.

1.5.2 Рекурентний перцептрон

На перший погляд схожий на звичайний перцептрон, єдина істотна відмінність полягає в тому, що його виходи потрапляють йому ж на входи, і беруть участь в обробці вже наступного вхідного вектора. Тобто, в разі поворотного перцептронну має місце не набір окремих, нічим не пов'язаних образів, а певний процес, і значення мають не тільки самі входи, але і те, в якій послідовності вони надходять.

Через це виникають відмінності в методі навчання - використовується те ж саме зворотне поширення помилки, але для того, щоб помилка потрапила по рекурентного зв'язку в минуле, використовуються різні хитрощі (якщо підійти до завдання «в лоб» - то виникне проблема догляду помилки на нескінченне число циклів назад). В іншому ж ситуація схожа на звичайний перцептрон - для навчання потрібно мати достатньо довгу послідовність пар вхід-вихід, яку потрібно багато раз прогнати через мережу, щоб її навчити (або ж мати під рукою мат. Модель шуканого процесу, яку можна ганяти у всіляких умовах, і в реалтайм давати результати мережі для навчання).

Мережа такого типу зазвичай добре вирішує завдання управління динамічними процесами (починаючи від класичної завдання стабілізації перевернутого маятника, і до будь-яких систем, якими взагалі хоч якось виходить управляти), передбачення динамічних процесів, крім курсу валют, і взагалі всього, де крім явно спостерігається входу у системи є деякий внутрішній стан, яке не зовсім зрозуміло як використовувати.

Сильні сторони: мережа дуже хороша для роботи з динамічними процесами.

Слабкі сторони: якщо все ж не працює, зрозуміти в чому проблема - дуже важко, в процесі навчання може вилетіти в самозбудження (коли сигнал, отриманий з виходу, забиває все, що приходить по входах), якщо рішення все ж отримано - складно зрозуміти, чи можна домогтися кращих результатів, і яким шляхом. Іншими словами, погано вивчена.

Перспективи: цей підхід явно не вичерпав себе в питаннях управління - власне, на даний момент рекурентні перцептрони використовуються досить рідко, хоча їх потенціал високий. Цікаві результати може дати підхід з безперервно адаптується до об'єкта управління мережею, хоча для цього ще необхідно вирішити проблему нестійкості навчання.

1.5.3 Згорткові нейронні мережі

Згорткові нейронні мережі - це багат шарові ІНС прямого поширення, що складаються з великої кількості перцептронів. У загальному випадку згорткова нейронна мережа складається з ланцюжка декількох згортальних шарів, які використовуються для виділення ознак, і декількох повно зв'язних шарів.

Запропонована Яном Лекуном в 1988 році, входить до складу технологій глибинного навчання. В основі згортальних нейронних мереж лежать такі ідеї:

- кожен нейрон шару отримує вхідний сигнал від локального рецептивного поля в попередньому шарі. Під рецептивних полем розуміється шар нейронів, що сприймають зовнішні сигнали і передають їх наступного шару.

При цьому рецептивне поле безпосередньо взаємодіє з наступним шаром;

- кожен прихований шар нейронної мережі складається з безлічі карт ознак (feature maps). У них все нейрони мають загальні ваги;

- за кожним шаром згортки слід обчислювальний шар, який здійснює локальне усереднення карт і просторову підвибірку.

При обчисленні мережі виходить, що кожен нейрон виконує операцію згортки (Конволюція) деякої області попереднього шару, яка визначається безліччю нейронів, пов'язаних з даними нейроном. Шари нейронної мережі, побудовані таким чином, називаються згортковими шарами. Крім, згортальних шарів в згортковій нейронної мережі можуть бути шари

субдискретизація і повно зв'язні шари. Шари субдискретизація виконують функції зменшення розмірності простору карт ознак. Даний шар дозволяє прискорити подальші обчислення. субдискретизація можливо завдяки тому, що в цій архітектурі нейронних мереж важливо не стільки значення ознаки, скільки сам факт його наявності. Вихідний шар згорткової нейронної мережі, як правило, завжди повно зв'язну. Нейрони цього шару мають повне з'єднання з усіма активації попереднього шару. Всі три види шарів можуть чергуватися в довільному порядку. Це дозволяє складати карти ознак з карт ознак, що на практиці означає здатність розпізнавання складних ієрархій ознак. Це дозволяє від конкретних особливостей вхідних даних до абстрактних деталей.

При навчанні нейронних мереж зазвичай використовується метод зворотного поширення помилки (Backpropagation) або його модифікації. Суть даного методу полягає в тому, що поширення сигналів помилки в ІНС походить від виходів мережі до її входів, в напрямку, зворотному прямому поширенню сигналів. Даний вид нейронних мереж дуже добре зарекомендував себе для вирішення безлічі завдань обробки зображень [3].

Однією з найбільш відомих моделей згортальних мереж, що застосовуються для вирішення завдання збільшення зображень, є розроблена в 2014 році модель нейронної мережі SRCNN (Super-Resolution Convolutional Neural Network) [14]. На вхід SRCNN подається зображення, збільшене за допомогою методів інтерполяції, яке потім проходить процес поліпшення якості. Подібний метод застосовується в розробленому компанією Google методі RAISR [27]. Існують безліч модифікацій даної моделі, одними з найшвидших за швидкістю і ефективних за якістю є VDSR (Very Deep Convolutional Network) [14] і FSRCNN (Fast Super-Resolution Convolutional Neural Network) [9]. Також заслуговує на увагу модель ESPCN (Efficient Sub-Pixel Convolutional Neural Network) [18], яка здатна в реальному часі збільшувати зображення. На відміну від SRCNN, на вхід подається не 18 збільшене за допомогою інтерполяції зображення, а зображення низького

дозвіл. Збільшення розмірів зображення відбувається в результаті роботи нейронної мережі.

1.5.4 Мережа радіально-базисних функцій

Нейронна мережа прямого поширення сигналу, яка містить проміжний (прихований) шар радіально симетричних нейронів. Такий нейрон перетворює відстань від даного вхідного вектору до відповідного йому "центру" по деякому нелінійному закону (зазвичай функція Гаусса) [34]. Мережа радіально-базисних функцій зображена на рисунку 1.2.

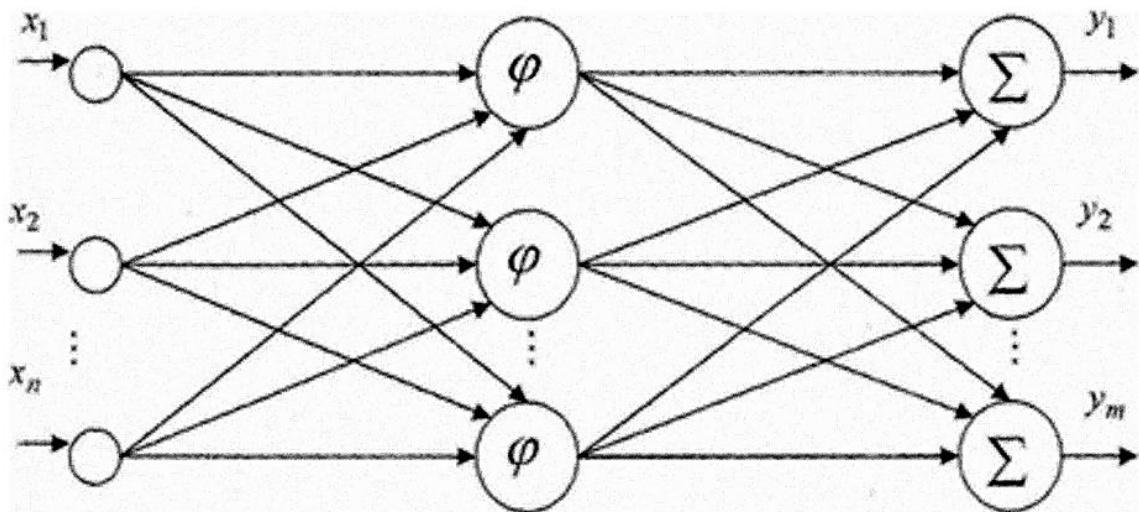


Рисунок 1.2 - Мережа радіально-базисних функцій

Радіально-базисна мережа характеризується трьома особливостями:

- єдиний прихований шар;
- тільки нейрони прихованого шару мають нелінійну активаційну функцію;
- синаптичні ваги зв'язків вхідного і прихованого шарів дорівнюють одиниці.

1.5.5 Повно зв'язні нейронні мережі

Повно зв'язні нейронні мережі мають шарувату будову, при якому кожен нейрон поточного шару має зв'язки з усіма нейронами попереднього шару. Мережа складається з довільного числа шарів. Розрізняють вхідний шар - на який надходить вектор (зазвичай він не враховується у підрахунку шарів), вихідного шару, який видає результат розрахунків мережі і прихованих шарів. Вхідний шар не виконує ніяких обчислень - він просто розподіляє вектор на наступний шар і служить приймачем інформації. Зазвичай, якщо прихованих шарів більше, ніж два, то у всіх прихованих шарах однакову кількість нейронів.

Кожен шар розраховує нелінійне перетворення від лінійної комбінації сигналів попереднього шару. Багатошарова нейронна мережа може формувати на виході довільну багатовимірну функцію при відповідному виборі кількості шарів, діапазону зміни сигналів і параметрів нейронів. Багатошарові нейронні мережі є універсальним аппроксиматором функцій [28]. Даний тип мереж здатний успішно вирішувати широкий спектр завдань і широко застосовується через свою простоту.

Однак, застосовні вони не до всіх даних: повно зв'язні мережі несприйнятливі до інваріантним даними і не здатні розпізнавати образи при різному їх розташуванні вхідного вектора. Складність навчання даної моделі полягає в тому, що, по-перше, необхідно навчати відразу дві нейронних мережі, а по-друге, необхідно правильно налаштувати баланс між ними. Генератор навчається створювати більш якісний результат, дискримінатор навчається розпізнавати реальні дані від згенерованих. Так само можливості даного виду мереж дуже сильно залежать від кількості шарів: при наявності менше трьох шарів мережа не здатна до класифікації нелінійна розділених об'єктів. Топологія мережі розташована на рисунку 1.3.

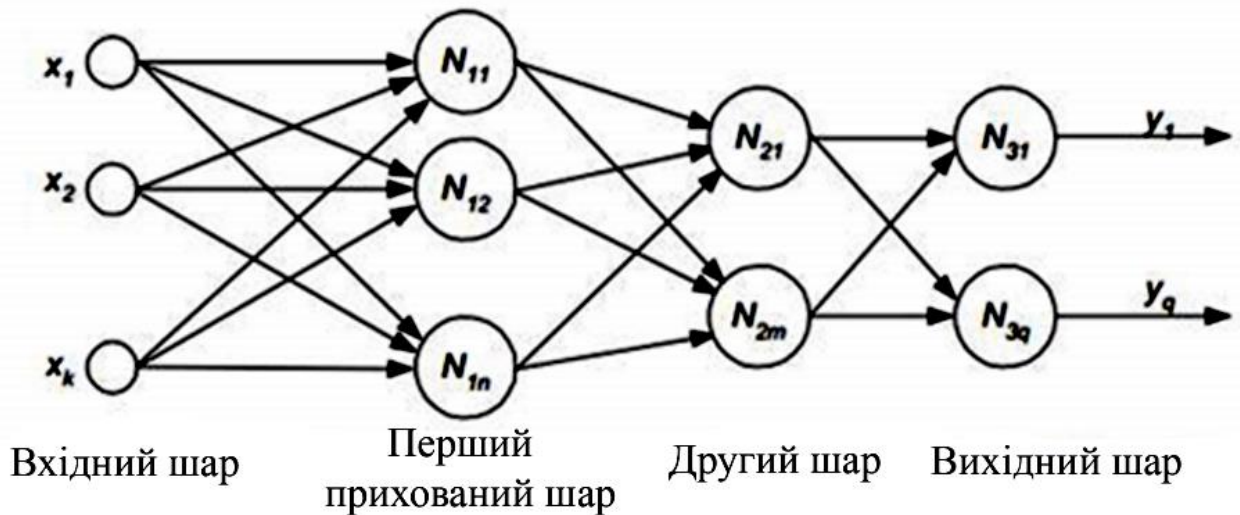


Рисунок 1.3 - Мережа прямого поширення сигналу

1.5.6 Генеративно-змагальні мережі

Генеративно-змагальні мережі - це алгоритм машинного навчання без учителя, побудований на комбінації з двох нейронних мереж, як правило згортальних або прямого поширення

Перша мережа (генератор) генерує зразки. Друга мережа (дискриміратор) приймає на вхід результати роботи першої мережі і правильні («справжні») дані, а потім намагається відрізнити справжні зразки від згенерованих. Вперше ця модель була описана співробітниками компанії Google в 2014 році [12].

Складність навчання даної моделі полягає в тому, що, по-перше, необхідно навчати відразу дві нейронних мережі, а по-друге, необхідно правильно налаштувати баланс між ними. Генератор навчається створювати більш якісний результат, дискриміратор навчається розпізнавати реальні дані від згенерованих.

Якщо одна з мереж буде навчена краще, ніж інша, то модель в цілому не буде працювати коректно. Модель навчається методом зворотного поширення помилки.

Для навчання зазвичай використовується наступний алгоритм:

- проводиться підготовка реальних даних. У задачі збільшення зображень - це вихідні зображення, до яких не було застосовано масштабування;
- генерується шум, на базі якого генератор генерує дані;
- формується набір даних для навчання дискримінатора. Цей набір який складається зі справжніх даних, підготовлених на кроці 1 (їм присвоюється мітка 1) і підробок від генератора, отриманих на попередньому кроці (їм присвоюється мітка 0);
- проводиться навчання дискримінатора, в процесі якого на вхід подаються реальні дані і створені генератором;
- проводиться навчання генератора. На даному етапі навчається генератор, а навчання дискримінатора відключено. На вхід подається шум, на виході очікується отримання позначки 1. Розпізнавання дискримінатором даних як реальних. При навчанні генератора не використовуються реальні зображення, а використовується тільки відмітка дискримінатора. Схема моделі змагальної мережі представлена на рисунку 1.4.

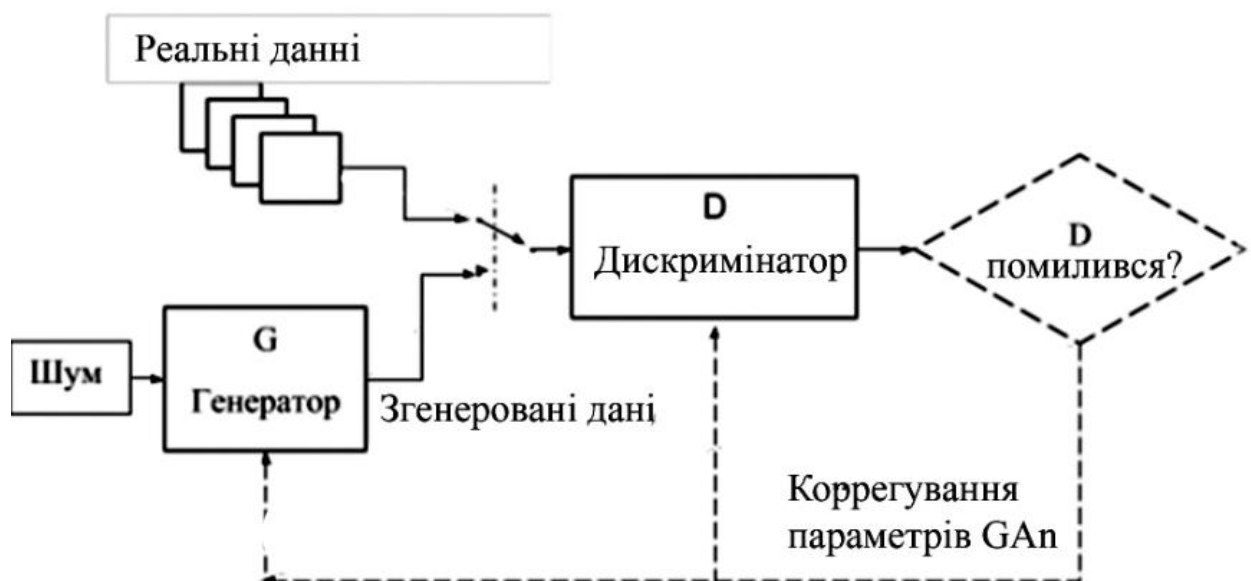


Рисунок 1.4 - Модель генеративно-змагальної нейронної мережі [22].

На даний момент однією з останніх моделей, розроблених для вирішення завдання збільшення дозволу зображення за допомогою ІНС, є SRGAN (Super-Resolution Generative Adversarial Network) [16].

1.5.7 Мережа Хопфілда

Повно зв'язна нейронна мережа із симетричною матрицею зв'язків. В процесі роботи динаміка таких мереж сходиться до одного з положень рівноваги. Ці положення рівноваги визначаються заздалегідь в процесі навчання, вони є локальними мінімумами функціонала, званого енергією мережі (в найпростішому випадку - локальними мінімумами негативно певної квадратичної форми на n -вимірному кубі).

Така мережа може бути використана як автоасоціативна пам'ять, як фільтр, а також для вирішення деяких завдань оптимізації. На відміну від багатьох нейронних мереж, що працюють до отримання відповіді через певну кількість тактів, мережі Хопфілда працюють до досягнення рівноваги, коли наступний стан мережі в точності дорівнює попередньому: початковий стан є вхідним чином, а при рівновазі отримують вихідний образ [36]. Мережа Хопфілда зображена на рисунку 1.5.

Мережа Хопфілда одношарова і складається з N штучних нейронів. Кожен нейрон системи може приймати на вході і на виході одне з двох станів (що аналогічно виходу нейрона з пороговою функцією активації).

Мережа складається з довільного числа шарів. Розрізняють вхідний шар - на який надходить вектор (зазвичай він не враховується у підрахунку шарів), вихідного шару, який видає результат розрахунків мережі і прихованих шарів. Вхідний шар не виконує ніяких обчислень - він просто розподіляє вектор на наступний шар і служить приймачем інформації. Зазвичай, якщо прихованих шарів більше, ніж два, то у всіх прихованих шарах однаково кількість нейронів.

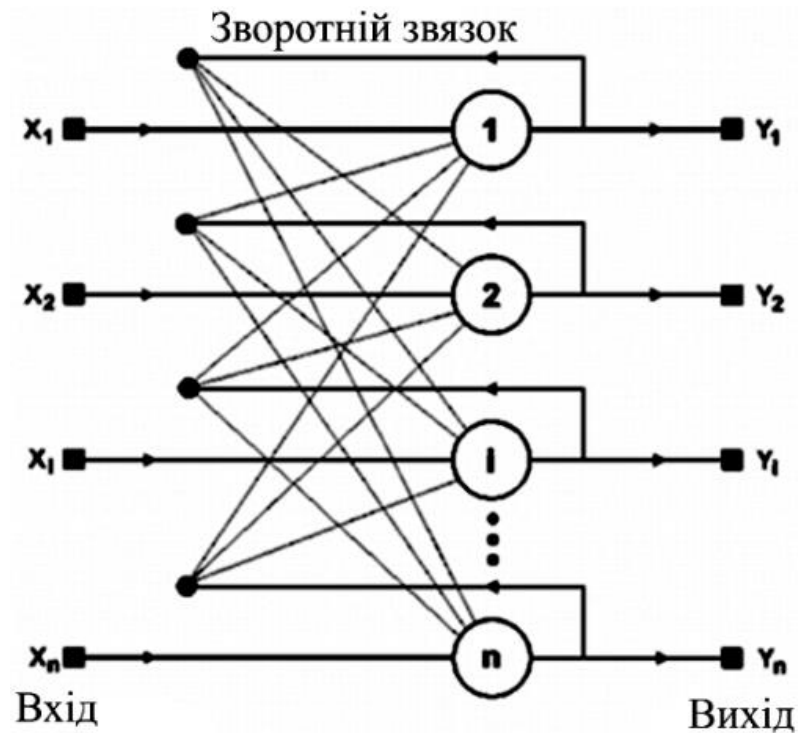


Рисунок 1.5 - Структурна схема мережі Хопфілда

1.5.8 LSTM И GRU

Мережі LSTM і GRU позбавлені недоліків мереж Джордана і Елмана і здатні встановлювати кореляцію так само між віддаленими одна від одної об'єктами. Обумовлено це більш складною архітектурою і наявністю додаткових шарів. Архітектура LSTM представлена на рисунку 1.6. Довга короткострокова пам'ять - різновид архітектури рекурентних нейронних мереж, запропонована в 1997 році Сеппі Хохрайтером і Юргеном Шмідхубер. Як і більшість рекурентних нейронних мереж, LSTM-мережа є універсальною в тому сенсі, що при достатній кількості елементів мережі вона може виконати будь-яке обчислення, на яке здатний звичайний комп'ютер, для чого необхідна відповідна матриця ваг, яка може розглядатися як програма. На відміну від традиційних рекурентних нейронних мереж, LSTM-мережу добре пристосовано до навчання на задачах класифікації, обробки і прогнозування часових рядів у випадках, коли важливі події розділені часовими лагами з

невизначеною тривалістю і кордонами. Відносна несприйнятливість до тривалості тимчасових розривів дає LSTM перевага по відношенню до альтернативних рекурентним нейронних мереж, прихованим марковським моделям і іншим методам навчання для послідовностей в різних сферах застосування [41]. Структура мережі LSTM зображена на рисунку 1.6.

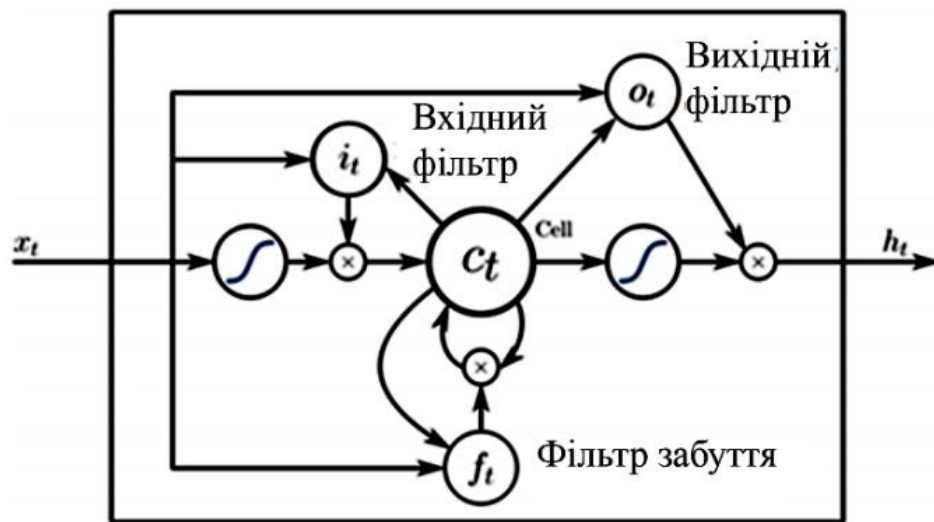


Рисунок 1.6 - Структура мережі LSTM

Мережа GRU відрізняється невеликою зміною зв'язку і відсутністю вихідного фільтра, що знижує кількість обчислення і параметрів, необхідних для роботи мережі. Однак це не позначається на точності роботи мережі [42].

1.5.9 Мережі Джордана і Елма

Мережі Джордана і Елмана є модифікацією багатошарової повномережі, отриманих за допомогою додавання шару контексту в топологію.

Однак, відрізняються вони тим, що в мережі Елмана шар контексту пов'язаний з прихованим шаром [39], а в мережі Джордана шар контексту пов'язаний з вихідним шаром [40]. Мережі даного типу можна побачити на рисунку 1.7.

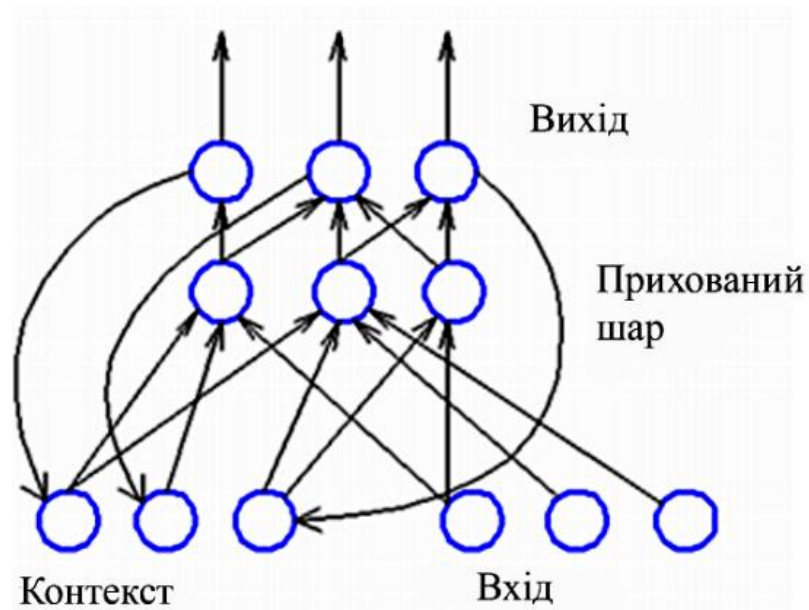


Рисунок 1.7 - Мережа Джордана

Однак, якщо затримка між значущими подіями стає великою, то мережі даного типу можуть працювати некоректно.

1.5.10 Само організаційна карта Кохонена

Нейронна мережа з навчанням без учителя, що виконує завдання візуалізації і кластеризації. Само організована карта складається з компонентів, які називаються вузлами або нейронами. Їх кількість задається аналітиком. Кожен з вузлів описується двома векторами. Перший - т. Зв. вектор ваги m , що має таку ж розмірність, що і вхідні дані. Другий - вектор r , що представляє собою координати вузла на карті.

Карта Кохонена візуально відображається за допомогою осередків прямокутної або шестикутної форми; остання застосовується частіше, оскільки в цьому випадку відстані між центрами суміжних осередків однакові, що підвищує коректність візуалізації карти. Топологія самоорганізується карти Кохонена зображена на рисунку 1.8.

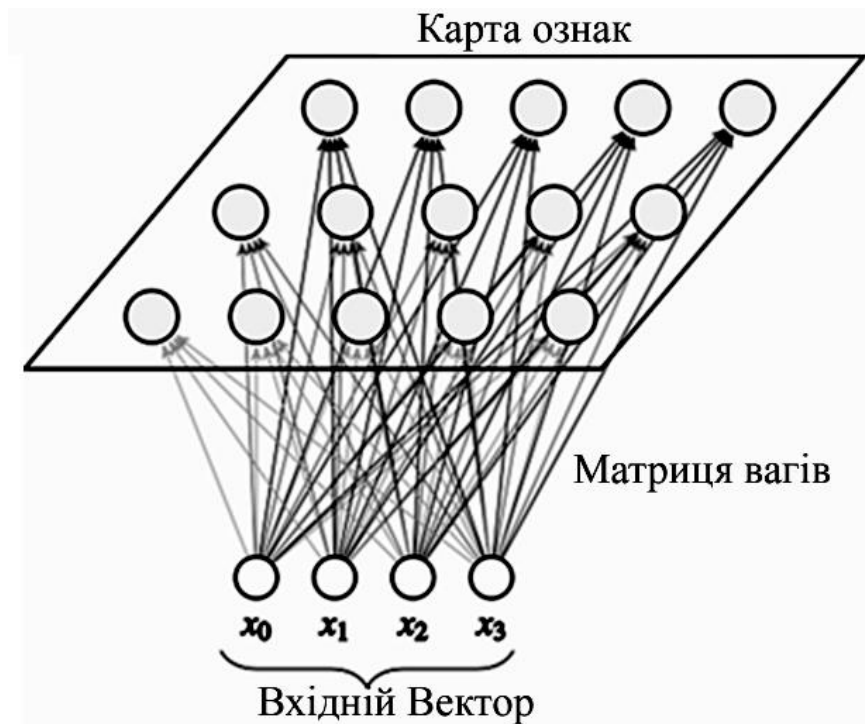


Рисунок 1.8 - Само організаційна карта Кохонена

Спочатку відома розмірність вхідних даних, по ній певним чином будується початковий варіант карти. В процесі навчання вектори ваги вузлів наближаються до вхідних даних. Для кожного спостереження (семпли) вибирається найбільш схожий по вектору ваги вузол, і значення його вектора ваги наближається до спостереження. Також до спостереження наближаються вектори ваги декількох вузлів, розташованих поруч, таким чином якщо в безлічі вхідних даних два спостереження були схожі, на карті їм будуть відповідати близькі вузли. Циклічний процес навчання, перебираючи вхідні дані, закінчується після досягнення картою допустимої (заздалегідь заданій аналітиком) похибки, або після здійснення заданої кількості ітерацій.

Таким чином, в результаті навчання карта Кохонена класифікує вхідні дані на кластери і візуально відображає багатовимірні вхідні дані в двовимірній площині, розподіляючи вектори близьких ознак в сусідні осередки і розфарбовуючи їх в залежності від аналізованих параметрів нейронів [38].

1.5.11 Мережа Хеммінга

Нейронна мережа Хеммінга - вид нейронної мережі, що використовується для класифікації бінарних векторів, основним критерієм в якій є відстань Хеммінга. Є розвитком нейронної мережі Хопфілда. Структура мережі Хеммінга зображена на рисунку 1.9.

Мережа Хеммінга - тришарова нейронна мережа зі зворотним зв'язком. Кількість нейронів у другому і третьому шарах дорівнює кількості класів класифікації.

Синапси нейронів другого шару з'єднані з кожним входом мережі, нейрони третього шару пов'язані між собою негативними зв'язками, крім синапсу, пов'язаного з власним аксоном кожного нейрона - він має позитивний зворотний зв'язок [37].

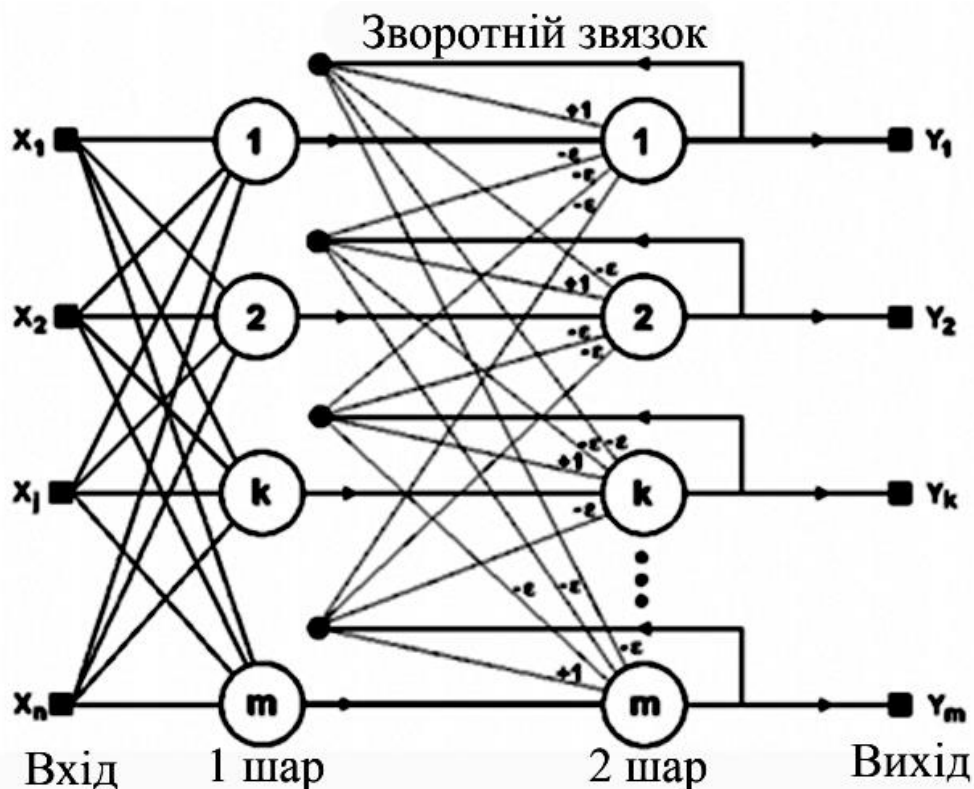


Рисунок 1.9 - Мережа Хеммінга

1.6 Постановка задачі дослідження

У цій атестаційній роботі ми розглянемо напрямок генерування зображень на основі генеративно-змагальних нейронних мереж. Ми описуємо методологію навчання генеративних змагальних мереж. Ключова ідея полягає в тому, щоб розвивати як генератор, так і дискримінатор поступово: починаючи з низької роздільної здатності ми додаємо нові шари, які моделюють все більш тонкі деталі навчання прогресує. Це одночасно прискорює тренування і значно стабілізує його, дозволяючи нам створювати зображення безпрецедентної якості, наприклад, зображення CELEBA.

Наше завдання навчити генеративно-змагальну нейронну мережу налаштувати мережеву архітектуру та ваги зв'язків для ефективного виконання різних завдань. Нейронна мережа повинна регулювати ваги зв'язку відповідно до наявного зразка тренувань. Функціонування мережі покращується в міру ітеративного регулювання ваг. Здатність мережі вчитися на прикладах робить їх більш привабливими порівняно із системами, які слідує певній системі правил роботи, сформульованій експертами.

Існують два концептуальних підходи до навчання нейронних мереж: навчання з вчителем і навчання без вчителя.

Навчання нейронної мережі з викладачем означає, що для кожного вхідного вектора навчального набору існує необхідне значення вихідного вектора, що називається цільовим. Ці вектори утворюють тренувальну пару. Ваги мережі змінюються до тих пір, поки не буде прийнятний рівень відхилення вихідного вектора від цілі для кожного вхідного вектора.

Навчання без вчителя має дещо інший характер і зустрічається рідше. За такого сценарію нейронна мережа не отримує бажаного результату. Ця підготовка підходить для мереж, які мають завдання кластеризації даних за заданими параметрами. Тобто, проаналізувавши велику кількість вхідних даних, мережа поділяє їх на категорії за певними характеристиками.

Підсилення підготовки використовується, коли можна оцінити чистий результат, отриманий мережею.

Тобто, заохочуючи нейронну мережу щоразу, коли результат буде максимально наближений до бажаного, ми дозволимо їй шукати будь-яке рішення проблеми до тих пір, поки воно дасть бажані результати.

Для проектування процесу навчання потрібно мати модель середовища, в якій працює нейронна мережа - знати інформацію, доступну для мережі. Вам також потрібно визначити, як змінювати вагу мережі - які правила навчання регулюють процес налаштування. Алгоритм тренування означає процедуру, яка використовує правила тренування для регулювання ваги.

Найважливішою особливістю нейронних мереж є їх здатність вчитися на даних про навколишнє середовище і, як результат, підвищувати свою продуктивність. Продуктивність збільшується з часом відповідно до певних правил. Тренування нейронної мережі здійснюється за допомогою інтерактивного процесу коригування синаптичних ваг і порогів. В ідеалі нейронна мережа отримує знання про навколишнє середовище при кожній ітерації навчального процесу.

Багато понять пов'язані з поняттям навчання, тому важко дати чітке визначення цьому процесу. Більше того, від цього залежить процес навчання. Це робить практично неможливим чітке визначення цього поняття. Наприклад, процес навчання з точки зору психолога принципово відрізняється від того, який має вчитель школи. З точки зору нейронної мережі, ймовірно, можна використовувати таке визначення.

Навчання - це процес, в якому налаштування вільних нейронних мереж налаштовуються за допомогою моделювання середовища, в яке вбудована ця мережа. Тип тренінгу визначається тим, як ці параметри коригуються.

Це визначення процесу навчання нейронної мережі передбачає таку послідовність подій:

- стимулюючи нейронну мережу, отримує подразники із зовнішнього середовища;

- внаслідок реалізації першого абзацу вільні параметри змінюються;
- після зміни внутрішньої структури нейронна мережа відповідає штрафи вже іншим чином.

Наведений вище перелік чітких правил вирішення проблеми навчання нейронної мережі називається алгоритмом навчання. Не існує універсального алгоритму навчання, який би підходив для всіх архітектур нейронної мережі. Існує лише набір інструментів, представлених багатьма алгоритмами навчання, кожен з яких має свої переваги. Алгоритми навчання відрізняються один від одного способом коригування ваги синаптичних нейронів. Ще однією чудовою особливістю є спосіб навчання нейронної мережі для з'єднання із зовнішнім світом. У цьому контексті вони говорять про парадигму навчання, пов'язану з моделлю середовища, в якій діє ця нейронна мережа.

Такий підхід до терміна "навчання нейронних мереж" узгоджується з біологічними нейронними мережами. Наш мозок складається з величезної кількості підключених нейронних мереж. Кожен складається з однотипних нейронів (функція активації однакова). Ми дізнаємось, змінюючи синапси - елементи, що підсилюють / послаблюють вхідний сигнал.

Однак є ще один важливий момент. Якщо ви тренуєте мережу, використовуючи лише один вхідний сигнал, мережа просто "запам'ятовує правильну відповідь". Ззовні, здавалося б, вона дуже швидко «навчилася». І як тільки ви подасте трохи змінений сигнал, чекаючи, щоб побачити правильну відповідь, мережа видасть нісенітницю.

Насправді, для чого нам потрібна мережа, яка визначає обличчя лише на одній фотографії. Ми очікуємо, що мережа зможе узагальнити деякі функції та розпізнати обличчя і на інших фотографіях.

Навчання нейронної мережі з учителем припускає, що для кожного вхідного вектора з навчальної множини існує необхідне значення вихідного вектора, званого цільовим.

Навчання нейронної мережі без вчителя є набагато більш правдоподібною моделлю навчання з точки зору біологічних коренів штучних нейронних мереж.

При змішаному навчанні частина ваг визначається за допомогою навчання з учителем, у той час як інша виходить за допомогою самонавчання.

Різні алгоритми навчання та пов'язані з ними архітектури мереж представлені у таблиці 1.1. В останній колонці перераховані задачі, для яких можуть бути застосовані різні алгоритми. Кожен з алгоритмів навчання орієнтований на мережу певної архітектури і призначений для обмеженого класу задач.

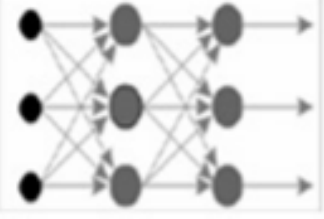

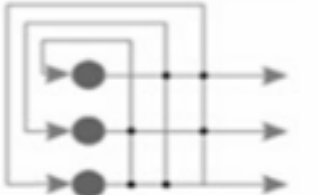
Таблиця 1.1 — Відомі алгоритми навчання нейронних мереж

Типи навчання	Правило навчання	Архітектура	Алгоритм навчання	Задача
З вчителем	Корекція помилки	Одношаровий та багатошаровий перцептрон	Алгоритми навчання перцептрона Зворотне поширення	Класифікація образів, Апроксимація функцій, Передбачення, управління
	Больцман	Рекурентна	Алгоритм навчання Больцмана	Класифікація образів
	Хебб	Багатошарова прямого розповсюдження	Лінійний дискримінантний аналіз	Аналіз даних, Класифікація образів
	Змагальні	Змагальня	Векторне квантування	Категоризація всередині класу, Аналіз даних
Мережа ART			ARTMap	Класифікація образів
Без вчителя	Корекція помилки	Багатошарова прямого розповсюдження	Проекція Саммона	Категоризація всередині класу, Аналіз даних
	Хебб	Прямого поширення або змагання	Аналіз головних компонентів	Аналіз даних, Стиснення даних
			Мережа Хапфілда	Навчання асоціативної пам'яті
	Змагальні	Змагальна	Векторне квантування	Категоризація, Стиснення даних

		Мережа Кохена	SOM Кохонена	Категоризація, Аналіз даних
		Мережа ART	ART1, ART2	Категоризація
Змішані	Корекція помилки та змагання	Мережа RBF	Алгоритм навчання RBF	Класифікація образів, Апроксимація функцій, Передбачення, Управління

ШНМ добре підходять для розпізнавання образів і вирішення задач класифікації, оптимізації і прогнозування. Зростання обсягів баз даних в техніці, бізнесі, медицині, екології і зростання вимог до точності рішення ставлять нові вимоги перед нейронними мережами. В таблиці 1.2 приведені типи ШНМ та класи задач, які вони вирішують.

Таблиця 1.2 – Типи штучних мереж

Типи ШНМ	Архітектура ШНМ	Клас задач які вирішує ШНМ
Багатошаровий перцептрон		Апроксимація функцій, класифікація
Само організаційна карта Кохонена		Стиснення даних, виділення признаков
Мережа Хапфілда		Асоціативна пам'ять, кластеризація даних

Одним із завдань, яку повинна виконувати система машинного бачення, є розпізнавання. Після оцінки різних властивостей об'єкта, наприклад відбивної здатності поверхні або кольору, система повинна мати у своєму розпорядженні деякою схемою, на підставі якої об'єкт буде віднесений до

одного з відомих класів. Класифікація образів забезпечує кошти для вирішення цього завдання. Основною ідеєю є виділення характеристик або ознак. Величин, які можуть виявитися корисними при розрізненні елементів різних класів. Значення окремих ознак потім об'єднуються і утворюють вектор ознак. Таким чином, інформація, отримана на основі зображення об'єкта, використовується для ідентифікації точки в деякому багатовимірному просторі.

Кластеризація (або кластерний аналіз) - це задача розбиття множини об'єктів на групи, які називаються кластерами. У середині кожної групи повинні виявитися «схожі» об'єкти, а об'єкти різних групи повинні бути якомога більш відмінні. Головна відмінність кластеризації від класифікації полягає в тому, що перелік груп чітко не заданий і визначається в процесі роботи алгоритму.

Апроксимація дозволяє досліджувати числові характеристики і якісні властивості об'єкта, зводячи завдання до вивчення більш простих або більш зручних об'єктів (наприклад, таких, характеристики яких легко обчислюються або властивості яких вже відомі). У теорії чисел вивчаються діофантови наближення, зокрема, наближення ірраціональних чисел раціональними. В геометрії розглядаються апроксимації кривих ламаними. Деякі розділи математики по суті цілком присвячені апроксимації, наприклад, теорія наближення функцій, чисельні методи аналізу.

Основне завдання апроксимації - побудова наближеної (апроксимуючої) функції, в цілому найближче проходить близько даних точок або близько цієї безперервної функції. Таке завдання виникає при наявності похибки у вихідних даних (в цьому випадку недоцільно проводити функцію точно через всі крапки, як в інтерполяції) або при бажанні отримати спрощене математичний опис складної або невідомої залежності.

Для початку потрібно скласти вектор характеристик для кожного об'єкта - як правило, це набір числових значень, наприклад, зростання-вага людини. Однак існують також алгоритми, що працюють з якісними (категорійними)

характеристиками. Після того, як ми визначили вектор характеристик, можна провести нормалізацію, щоб всі компоненти давали однаковий внесок при розрахунку «відстані».

У разі якщо апроксимація проводиться на безперервному безлічі точок (відрізку), апроксимація називається безперервної або інтегральної. Прикладом такої апроксимації може служити розкладання функції в ряд Тейлора, тобто заміна деякої функції статечним многочленом.

Найбільш часто зустрічають видом точкової апроксимації є інтерполяція - знаходження проміжних значень величини за наявним дискретному набору відомих значень.

Пам'ять, що адресується за змістом. У моделі обчислень фон Неймана звертання до пам'яті доступно тільки за допомогою адресу, який не залежить від утримання пам'яті. Більш того, якщо допущена помилка в обчисленні адреси, то може бути знайдена зовсім інша інформація. Асоціативна пам'ять, чи пам'ять, що адресується за змістом, доступна за вказівкою заданого змісту. Зміст пам'яті може бути викликаний навіть по частковому входу. Асоціативна пам'ять надзвичайно ефективна при створенні мультимедійних інформаційних баз даних.

Пам'ять нейронної мережі є відмово стійкої. При видаленні окремих частин нейронної мережі відбувається лише зниження якості інформації, в ній зберігається, але не повне її зникнення. Це відбувається тому, що інформація зберігається в розподіленій формі.

Програмні продукти на базі нейронних мереж використовують для контролю якості води та можуть знаходити пластикові бомби в багажі авіапасажирів. Фахівці інвестиційного банку за допомогою програмного нейро-пакета роблять короткострокові прогнози коливань курсів валют. Проведення аналізу вирішення завдань на нейронних мережах дозволяє виділити основні переваги їх використання: Рішення завдань при невідомих закономірностях. Використовуючи здатність навчання на безлічі прикладів, нейронна мережа здатна вирішувати завдання, в яких невідомі закономірності

розвитку ситуації і залежності між вхідними і вихідними даними. Традиційні математичні методи та експертні системи в таких випадках не використовують. Нейронні мережі дозволяють виконувати екстраполяцію і інтерполяцію на основі інформації, що зберігається в них. Тобто, навчання дозволяє надати мережі здатності здійснювати пошук важливих особливостей або зв'язків в даних. Після цього мережа в стані екстраполювати і виявляти зв'язки в нових даних, що до неї надходять. Наприклад, в одному експерименті було проведено навчання нейронної мережі на гіпотетичному прикладі. Після закінчення навчання мережу придбала здатність адекватно реагувати на питання, за якими навчання не проводилося. Для отримання нейронної мережі, що використовується для оцінки якісних характеристик проєктованих інформаційних систем, найбільш придатною є парадигма навчання з учителем. Такий підхід дозволяє врахувати думку експертів з проєктування і впровадження інформаційних рішень, а також дані про реальні проєкти, зібрані при їх аналізі.

У разі, коли прикладна задача не дозволяє створити повне безліч навчальних прикладів, яке задовольняє умові несуперечливості, пропонується використовувати нейронні нечіткі мережі. Цей підхід до навчання нейронної мережі може бути застосований при необхідності оцінки проєктів великих інформаційних систем з високою вартістю реалізації та впровадження.

Нейронні мережі мають потенційну надвисоку швидкодію за рахунок використання масового паралелізму обробки інформації. Нейронні мережі потенційно відмово стійкі при апаратній реалізації. Це означає, що за несприятливих умов їх продуктивність падає незначно. Дані в нейронній мережі зберігаються природним чином за допомогою асоціативної пам'яті. Асоціативної пам'яттю називають таку пам'ять, в якій досить виконати пошук частково представлених даних, щоб повністю відновити всю інформацію. У цьому полягає відмінність асоціативної пам'яті від звичайної пам'яті, в якій отримання даних здійснюється шляхом зазначення точної адреси відповідних елементів пам'яті.

2 СТРУКТУРА ТА МЕТОДИ ГЕНЕРАТИВНО-ЗМАГАЛЬНИХ НЕЙРОМЕРЕЖ

2.1 Структура генеративно-змагальних нейронних мереж

Генеративно-змагальна нейронна мережа (Generative adversarial network, GAN) - архітектура, що складається з генератора і дискримінатора, налаштованих на роботу один проти одного. Звідси GAN і отримала назву генеративно-змагальна. У разі роботи з зображеннями, в усьому іншому - це згорткові нейронні мережі.

2.1.1 Генератор

Генеративні алгоритми зайняті зворотним. Замість того, щоб передбачати категорію за наявними образами, вони намагаються підібрати образи до даної категорії.

Генеративна мережу накладає шум на зображення (використовують вектор шуму) і виводить його. Під час тренування генеративна мережа вивчає, які області зображення необхідно змінити або поліпшити, щоб дискримінатори знадобилося більше часу для визначення автентичності згенерованого зображення.

Генеративних мережу з кожним разом виробляє зображення, яке все більше схоже на реальне, в той час як дискримінативна мережа намагається знайти відмінності між реальним і штучним зображенням. Головна мета - зробити таку генеративну мережу, яка зможе відтворювати відрізнити від реальних зображення.

У той час як дискримінаційні алгоритми хвилює взаємозв'язок між y і x , генеративні алгоритми хвилює "звідки беруться x ". Вони дозволяють знаходити $p(x | y)$, ймовірність x при даному y або ймовірність образів при даному класі (генеративні алгоритми також можуть використовуватися в

якості класифікаторів. Вони можуть робити більше, ніж класифікувати вхідні дані.).

Ще одне уявлення про роботу генеративних алгоритмів можна отримати, розділяючи дискримінаційні моделі від генеративних таким чином:

- дискримінаційні моделі вивчають кордон між класами;
- генеративні моделі моделюють розподіл окремих класів.

2.1.2 Дискримінатор

Дискримінаційні алгоритми намагаються класифікувати вхідні дані. З огляду на особливості отриманих даних, вони намагаються визначити категорію, до якої вони належать.

Наприклад, пробігаючи всі слова в листі дискримінаційний алгоритм може передбачити, є повідомлення спамом або не спам. Спам - це категорія, а пакет слів, зібраний з електронної пошти - образи, які становлять вхідні дані. Математично категорії позначають y , а образи позначають x . Запис $p(y | x)$ використовується для позначення «ймовірності y при заданому x », яка позначає «ймовірність того, що електронний лист є спамом при наявному наборі слів». Мережа дискримінаторів є стандартною згортковою мережею, яка може класифікувати зображення, що подаються на неї за допомогою біноміального класифікатора, що розпізнає зображення як реальні або як підроблені. Генератор в певному сенсі є зворотний сверточное мережу: хоча стандартний згортковий класифікатор приймає зображення і зменшує його дозвіл, щоб отримати можливість, генератор приймає вектор випадкового шуму і перетворює його в зображення.

Отже, дискримінаційні функції зіставляють образи з категорією. Вони зайняті тільки цієї кореляцією.

2.1.3 Як працюють генеративно-змагальна нейронна мережа

Одна нейронна мережа, яка називається генератором, генерує нові екземпляри даних, а інша - дискримінатор, оцінює їх на справжність; тобто дискримінатор вирішує, чи належить кожен екземпляр даних, який він розглядає, до набору тренувальних даних чи ні. Перед тим як почати будувати дискримінатор і генератор, потрібно зібрати дані і зробити їх попередню обробку.

Припустимо, ми намагаємося зробити щось більш банальне, ніж повторити портрет Мони Лізи. Ми згенеруємо рукописні цифри, подібні до тих, що є в наборі даних MNIST. Мета дискримінатора - розпізнати справжні екземпляри з набору.

Тим часом, генератор створює нові зображення, які він передає дискримінатору. Він робить це в надії, що вони будуть прийняті справжніми, хоча є підробленими. Мета генератора полягає в тому, щоб генерувати рукописні цифри, які будуть пропущені дискримінатором. Мета дискримінатора - визначити, чи є зображення справжнім.

Кроки, які проходить GAN:

- генератор отримує випадкове число і повертає зображення;
- це згенеровані зображення подається в дискримінатор поряд з потоком зображень, взятих з фактичного набору даних;
- дискримінатор приймає як реальні, так і підроблені зображення і повертає ймовірності, числа від 0 до 1, причому 1 являє собою справжнє зображення і 0 представляє фальшиве.

Таким чином, у є подвійний цикл зворотного зв'язку:

- дискримінатор знаходиться в циклі з достовірними зображеннями;
- генератор знаходиться в циклі разом з дискримінатором.

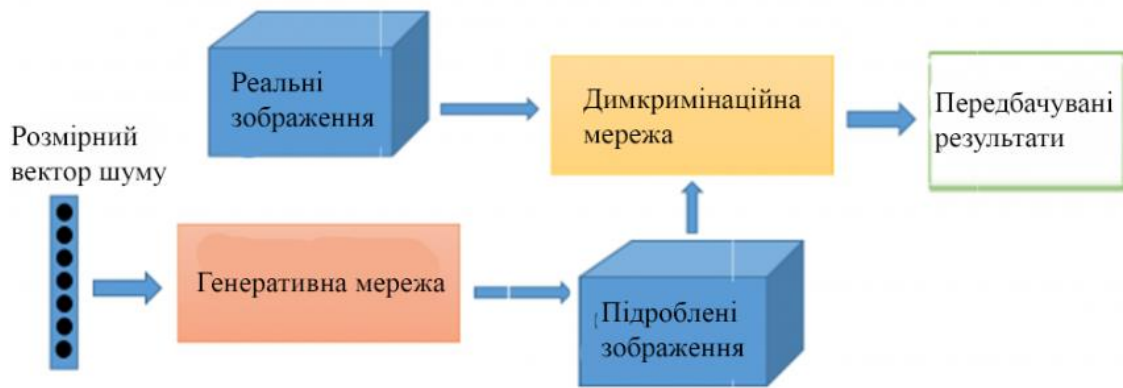


Рисунок 2.1 - Алгоритм роботи генеративно-змагальної нейронної мережі

Ви можете уявити GAN як фальшивомонетника і поліцейського, що грають в кішки мишки, де фальсифікатор вчиться виготовляти неправдиві купюри, а поліцейський вчиться їх виявляти. Обидва динамічні. поліцейський теж тренується (можливо, центральний банк відзначає пропущені купюри), і кожна сторона приходить до вивчення методів іншого в постійній ескалації.

Мережа дискримінаторів є стандартною згортковою мережею, яка може класифікувати зображення, що подаються на неї за допомогою біноміального класифікатора, що розпізнає зображення як реальні або як підроблені.

Генератор в певному сенсі є зворотний свертковий мережу: хоча стандартний згортковий класифікатор приймає зображення і зменшує його дозвіл, щоб отримати можливість, генератор приймає вектор випадкового шуму і перетворює його в зображення. Перший відсіває дані за допомогою методів зниження дискретизації, таких як maxpooling, а другий генерує нові дані.

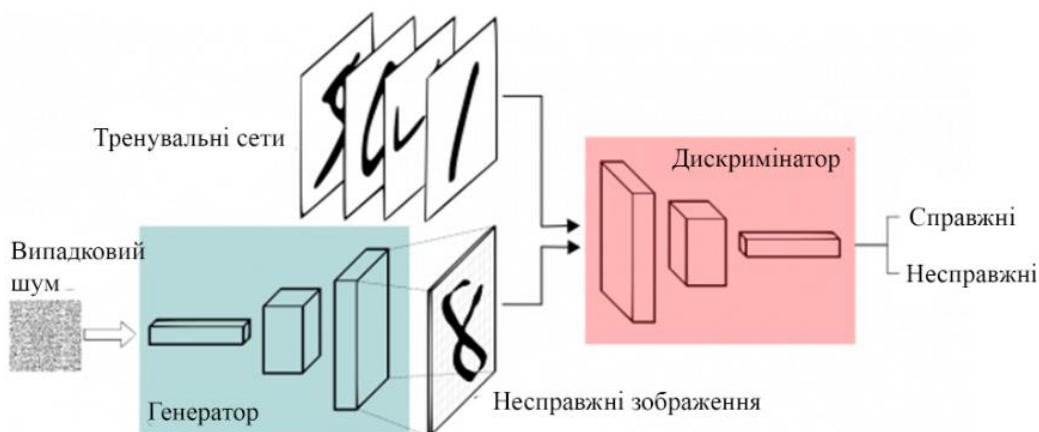


Рисунок 2.2 - Алгоритм роботи на основі генератора та дискримінатора

Обидві мережі намагаються оптимізувати цільову функцію або функцію втрат в грі zero-sum. Це, по суті, модель актора-критика (actor-critic). Коли дискримінатор змінює свою поведінку, то і генератор змінює, і навпаки.

2.1.4 Автокодери і VAE

Корисно порівняти генеративні змагальні мережі з іншими нейронними мережами, такими як автокодери (автоенкодери) і варіаційні автокодери.

Автокодери кодують вхідні дані в вектори. Вони створюють приховане або стислий уявлення необроблених даних. Вони корисні при зменшенні розмірності: вектор, службовець в якості прихованого уявлення, стискає необроблені дані в меншу кількість. Автокодери можуть бути пов'язані з так званим декодером, який дозволяє відновлювати вхідні дані на основі їх прихованого уявлення, як і у випадку з машиною Больцмана.

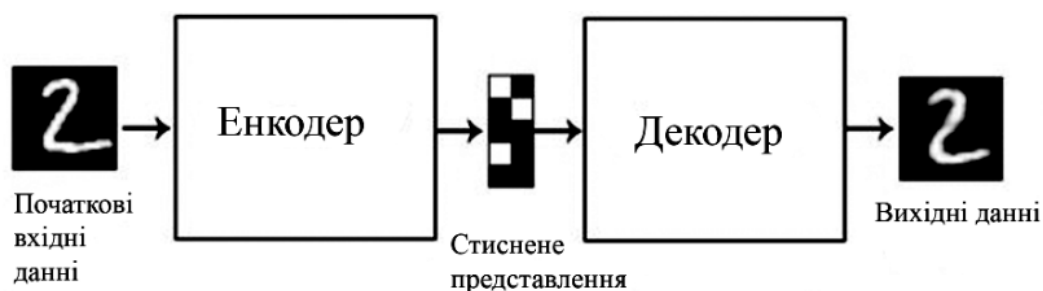


Рисунок 2.3 - Алгоритм роботи автокодерів

Варіаційні автокодери є генеративним алгоритмом, який додає додаткове обмеження для кодування вхідних даних, а саме те, що приховані уявлення нормалізуються. Варіаційні автокодери здатні стискати дані як автокодери і синтезувати дані подібно GAN. Однак, в той час як GAN генерують дані деталізовано, зображення, створені VAE, бувають більш розмитими. Приклади DeepLearning4j включають в себе як автокодери, так і варіаційні автокодери.

Ви можете розділити генеративні алгоритми на три типи, які маючи:

- категорію, пророкують пов'язані функції (Naive Bayes);
- приховане уявлення, пророкують пов'язані функції (VAE, GAN);
- деякі образи, пророкують інше (inpainting, imputation).

2.1.5 Поради з навчання GAN

Коли ви тренуєте дискримінатор, утримуйте значення генератора постійними; і коли ви тренуєте генератор, утримуйте дискримінатор на одному рівні. Кожен повинен тренуватися проти статичного противника. Наприклад, генератору це дозволить краще зчитувати градієнт, за яким він повинен вчитися.

Точно так же попереднє тренування дискримінатора проти MNIST сприятиме встановленню більш чіткого градієнта.

Кожна частина GAN може здолати іншу. Якщо дискримінатор дуже хороший, він буде повертати значення дуже близькі до 0 або до 1, так що генератор буде зазнавати труднощів у читанні градієнта. Якщо генератор дуже хороший, він буде постійно використовувати недоліки дискримінатора, що призводять до неправильних негативів.

GAN вимагають багато часу на тренування. На одному GPU тренування може займати години, а на одному CPU - більше одного дня.

Незважаючи на складність настройки і, отже, використання, GAN стимулювали створення багатьох цікавих досліджень і статей, наприклад:

- PGGAN - прогресивна генеративна нейронна мережа від Nvidia;
- TwinGAN - міждоменої перенесення людських портретів;
- XOGAN: перетворення одного зображення в кілька;
- метод відновлення очей за допомогою Exemplar GAN від Facebook AI Research.

2.2 Проблеми тренування

Генеративно-змагальні мережі сумно відомі своєю складністю в підготовці і нестабільністю роботи. Одна з проблем полягає в тому, що якщо генератор занадто сильно випереджає дискримінатор в темпах навчання, то його вибірка зображень звужується до конкретно тих, які допомагають йому обдурити дискримінатор. Фактично, в результаті навчання генератора зводиться до створення одного-єдиного універсального зображення для обману дискримінатора. Ця проблема називається «режимом колапсу».



Рисунок 2.4 - Зображення згенеровані генеративно-змагальною нейронною мережею

Режим колапсу генеративно-змагальних мереж схожий на Goodfellow's. Зверніть увагу, багато хто з цих зображень спальні виглядають дуже схожими один на одного.

Інша проблема полягає в тому, що, коли генератор ефективно обманює дискримінатор оперує дуже маленьким градієнтом, тому не може отримати достатньо даних для знаходження істинного 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 з простих змін, запропонованих Xudong Mao et al., 2016, є заміна функції втрат на пару простих функцій, в основі яких лежать квадрати меншої площі. Це призводить до стабілізації процесу тренування, отриманню більш якісних зображень і меншому шансу колапсу з використанням незатухаючих градієнтів.

Інша проблема, з якою зіткнулися дослідники, - це складність отримання зображень з високою роздільною здатністю, почасти через те, що більш детальне зображення дає дискримінатори більше інформації для виявлення підроблених зображень. Сучасні генеративно-змагальні мережі починають навчати мережу з картинок в низькій роздільній здатності і поступово додають все більше і більше шарів, поки не буде досягнутий бажаний розмір зображення.

Поступове додавання шарів з більш високою роздільною здатністю під час навчання генеративно-змагальних мереж істотно підвищує стабільність усього процесу, а також швидкість і якість одержуваного зображення.

2.3 Як і де застосовуються GAN і генеративні моделі

Створення контенту і даних: наприклад, створення картинок для інтернет-магазину, аватарів для ігор, відеокліпів, згенерованих автоматично, виходячи з музичного біта твору, або навіть віртуальних провідних для ТВ-програм. Завдяки роботі GAN і генеративних моделей виникає синтез даних, на яких потім будуть навчатися інші системи.

Автоматичне редагування: цей підхід вже використовується в сучасних смартфонах і деяких програмах. Він дозволяє змінювати на фотографії обличчя, кількість зморшок і зачіску, міняти день на ніч, старіють зображення і т.п.

Генерація приголомшливо реалістичних зображень "знаменитостей" (а насправді - неіснуючих людей) за допомогою PG GAN від nVidia. С допомогою цієї мережі можна також генерувати зображення будь-яких інших категорій.



Рисунок 2.5 - Зображення згенерованих обличч знаменитостей

Всі ці зображення згенеровані системою на базі генеративно-змагальних нейронних мереж, частина з них виглядає не дуже реалістично, але інша частина вельми правдоподібна.



Рисунок 2.6 - Зображення згенеровані ГЗМ

Модель Everybody Dance Now, створена групою вчених з Університету Берклі, представляє простий метод на основі генеративних нейронних мереж передачі руху «зроби як я». Дано: вихідне відео танцюючої людини. Досягнутий результат: ми змушуємо зображення іншої людини танцювати. Це називається "перенесення руху".



Рисунок 2.7 - Кадр з відео прикладу роботи Everybody Dance Now

Перенесення стилю з одного зображення на інше дозволяє за допомогою неймереж робити такі вражаючі речі як "перетворити коня в зебру "



Рисунок 2.8 - Приклади підміни одного зображення на інше

Або генерувати "аніме-портрети" по фотографії. На цій картинці видно, як з цим завданням справляються різні види Ганов.

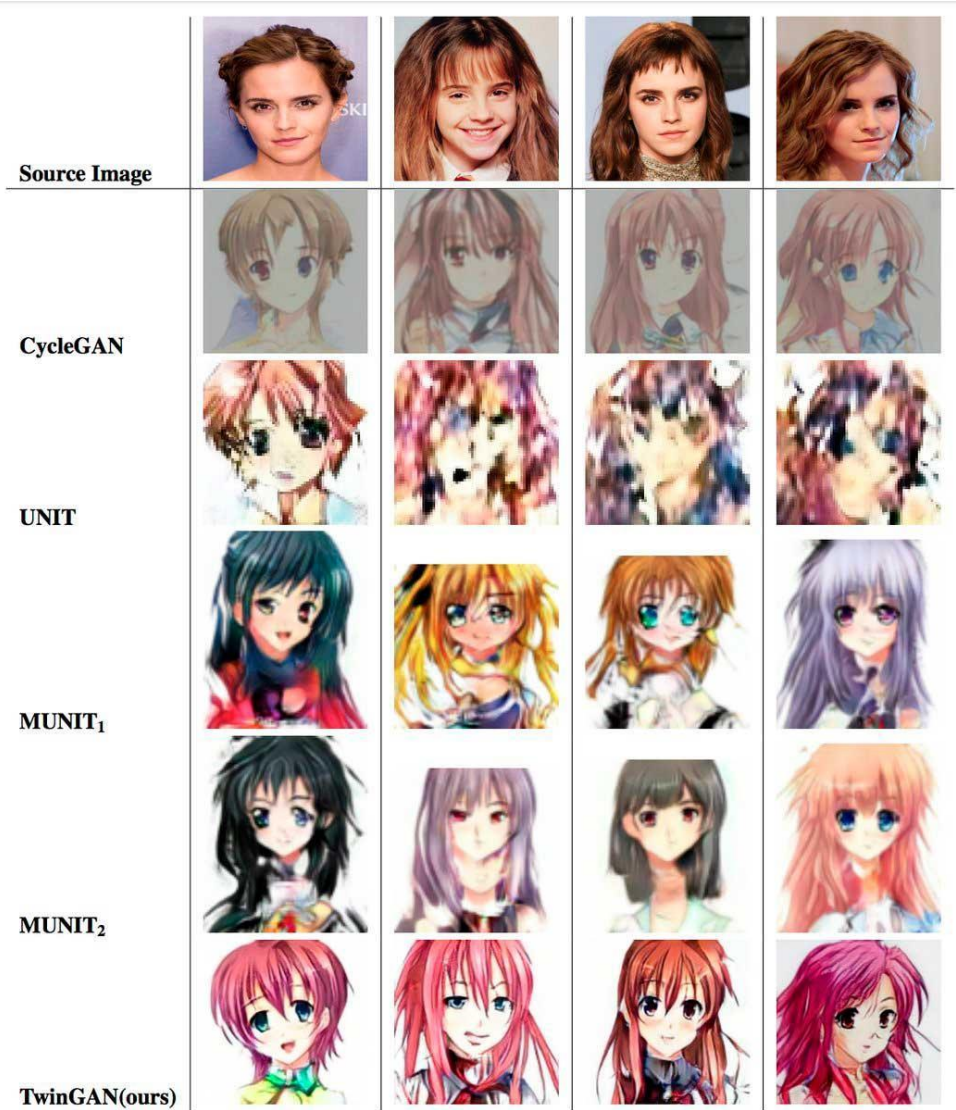


Рисунок 2.9 - Приклад згенерованих аніме обличь

Зміна емоцій, віку, виразу обличчя людини - всього цього можна досягти, правильно навчивши і запрограмувавши нейронні мережі GAN. На практиці це виглядає наступним чином: на вході моделі подається вихідне фото і вказується емоція, яку потрібно показати на виході.



Рисунок 2.10 - Приклад підміни емоцій на обличчі

Також нейронні мережі GAN використовують для генерації реалістичного відео міського середовища. Наприклад, при створенні фільмів, ігор, віртуальної реальності.

Перетворення начерків і контурних малюнків в фото реалістичне зображення за допомогою GAN відбувається наступним чином: ви малюєте від руки особа, сумку або, наприклад, kota і отримуєте фото реалістичне зображення на виході.

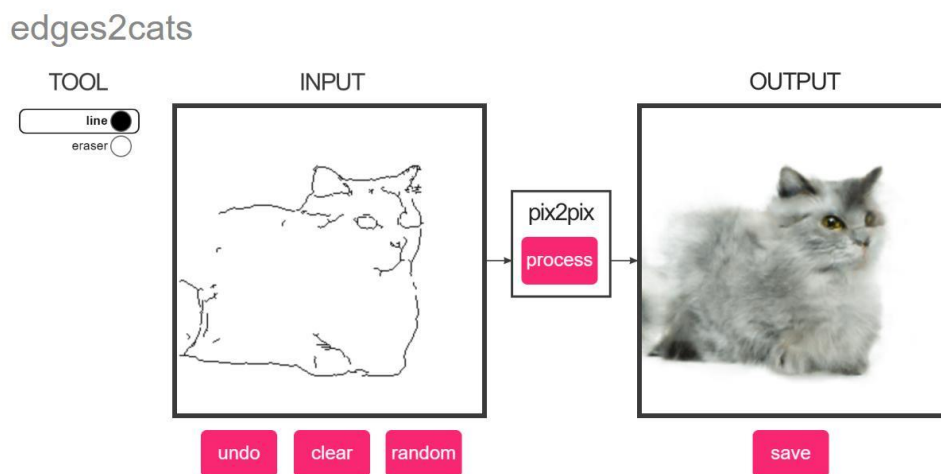


Рисунок 2.11 - Генерація зображення kota по малюнку від руки

Оскільки GAN - досить молодий метод (він з'явився тільки в 2014 році, а перші "пристойні" результати з'явилися в 2016-2017 роках), ще ніхто не знає достовірно, що можна, а чого не можна зробити за допомогою GAN. Оглядам нових варіацій GAN і способам їх застосування присвячено багато статей, ми ж зберемо найцікавіші з них і будемо публікувати їх в нашому блозі.

Для чого ми в Evergreen плануємо застосовувати нейронні мережі GAN? Основна ідея - автоматизація виробництва контенту, який раніше вимагав зусиль дизайнерів.

Якщо ви вважаєте, що зараз переплачуєте за створення контенту, який може створювати машина, поділіться з нами своєю ідеєю. Ми будемо раді знайти метод генерації, відповідний вашим потребам.

2.4 Генеративно-змагальна нейронна мережа Spade

Ідея SPADE полягає в тому, щоб запобігти втраті семантичної інформації в мережі, дозволяючи мапі сегментації управляти 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 дискретизацій через 2 згортальних шару.

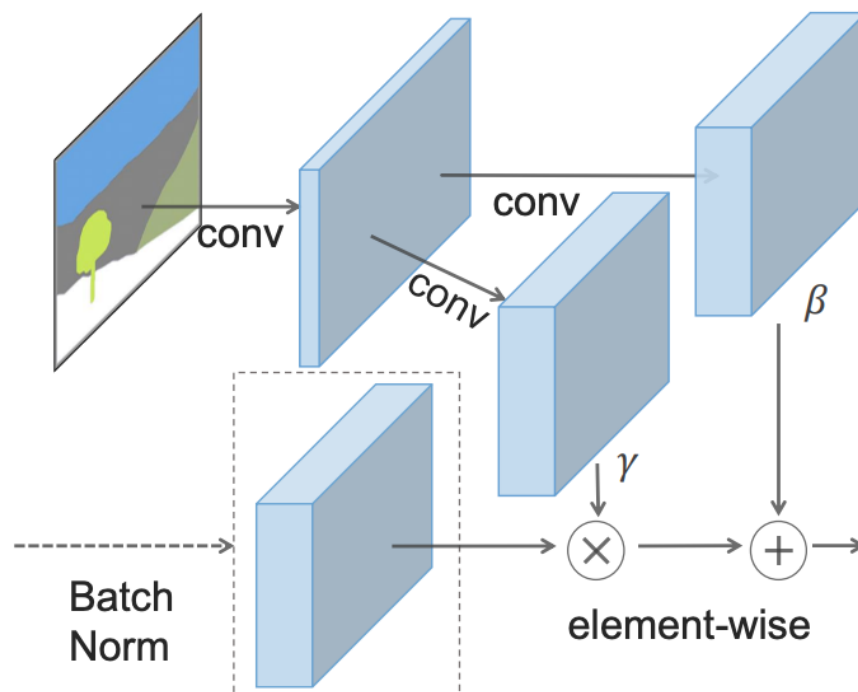


Рисунок 2.12 - Алгоритм роботи ГЗМ SPADE

Замість того щоб розгортати карту сегментації на перший шар, SPADE використовує її версії зі зниженою дискретизацій для модуляції нормалізованих вихідних даних для кожного шару. Генератор SPADE об'єднує всю цю конструкцію в невеликі «залишкові блоки», які містяться між шарами підвищує дискретизації (транспонована згортка):

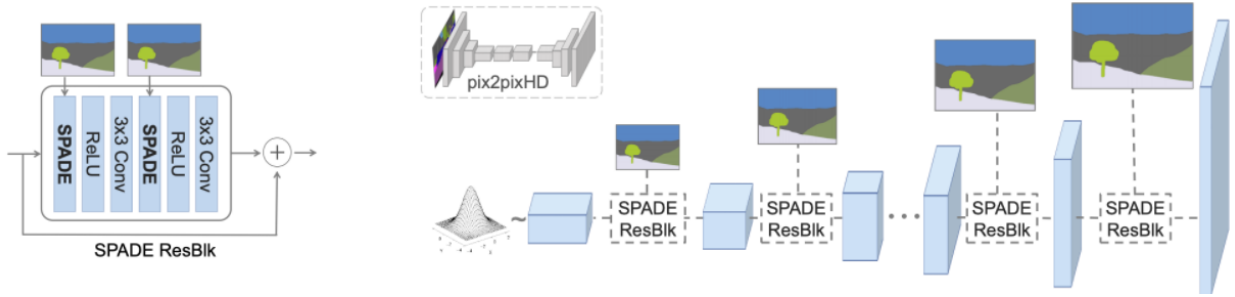


Рисунок 2.13 - Схема роботи ГЗМ SPADE

Високорівнева схема генератора SPADE в порівнянні з генератором pix2pixHD.

Тепер, коли карта сегментації подається «зсередини» мережі, немає необхідності використовувати її в якості вхідних даних для першого шару. Замість цього ми можемо повернутися до вихідної схеми генеративно-змагальної мережі, в якій в якості вхідних даних 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 функцію втрат, що і pix2pixHD, але з однією зміною: замість зведення в квадрат значень VLSGAN в ній використовується hinge loss.

З цими змінами ми отримуємо прекрасні результати:

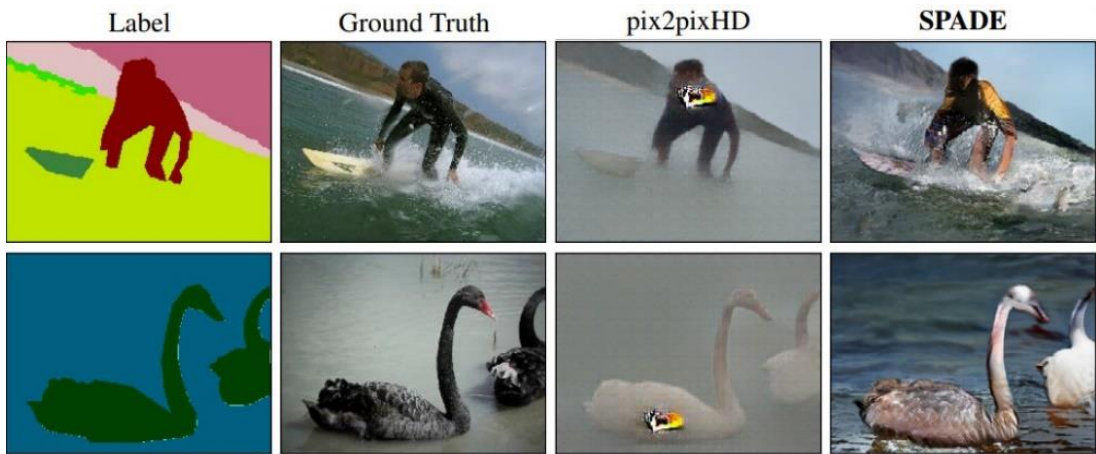


Рисунок 2.14 - Результати SPADE порівнюються з результатами pix2pixHD

2.5 Висновок і подальші дослідження

З наведеного аналізу видно що всі класичні генеративно-змагальні мережі (за винятком Wasserstein GAN, яка має свої способи поліпшення стабільності) мають «хорошими» полями. Тобто дотримання цих полів має єдину точку покаятися в якій розподіл генератора одно розподілу даних.

Чому ж тоді навчання генеративно-змагальних мереж така важка задача. Відповідь проста - параметризація нейронних мереж. При «поганий» параметризації ми можемо так само піти гуляти колами. Наприклад в мої експерименти показують що, наприклад, використання BatchNormalization в кожній з мереж відразу перетворює поле в замкнутий. А найкраще працює Relu активація.

На жаль на даний момент немає жодного способу теоретично перевірити які елементи нейромереж як змінюють поле. Мені доведе перспективний досліджувати властивості параметричного ядра системи - $K_{\theta}(x, y)$.

3 КОМП'ЮТЕРНА МОДЕЛЬ ГЕНЕРАЦІЇ ЗОБРАЖЕНЬ

3.1 Прогресивне навчання генеративно-змагальної мережі

Наш головний внесок - це методика навчання генеративно-змагальної мережі, де ми починаємо з низької роздільної здатності зображення, а потім поступово збільшуєте роздільну здатність, додаючи шари до мереж у вигляді візуалізації на рис. 1. Цей поступовий характер дозволяє навчанням спочатку виявити масштабну структуру розподіл зображення, а потім перевести увагу на дедалі точніші деталі масштабу, замість того, щоб вчитися всі масштаби одночасно

Ми використовуємо генераторні та дискримінаційні мережі, які є дзеркальними зображеннями один одного і завжди зростають синхронно. Усі існуючі шари в обох мережах залишаються відстежуваний протягом навчального процесу. Коли нові шари додаються до мереж, ми плавно їх вицвітають, як Проілюстровано на рисунку 3.1. Це дозволяє уникнути раптових потрясінь уже добре навчених шарів меншої роздільної здатності. У додатку А детально описана структура генератора та дискримінатора, а також інші тренінги параметри.

Ми спостерігаємо, що прогресивне навчання має ряд переваг. На початку покоління менших зображень значно стабільніше, оскільки є менше інформації про клас та менше режимів (Odena) та ін., 2017). Поступово збільшуючи роздільну здатність, ми постійно задаємо набагато простіше питання порівняно з кінцевою метою виявлення відображення від прихованих векторів до напр. 10242 образи. Цей підхід має концептуальну схожість з недавньою роботою Chen & Koltun (2017). На практиці це достатньо стабілізує навчання для надійного синтезування мега піксельних зображень за допомогою втрат WGAN-GP (Gulrajani et al., 2017) і навіть втрати LSGAN (Mao et al., 2016b).

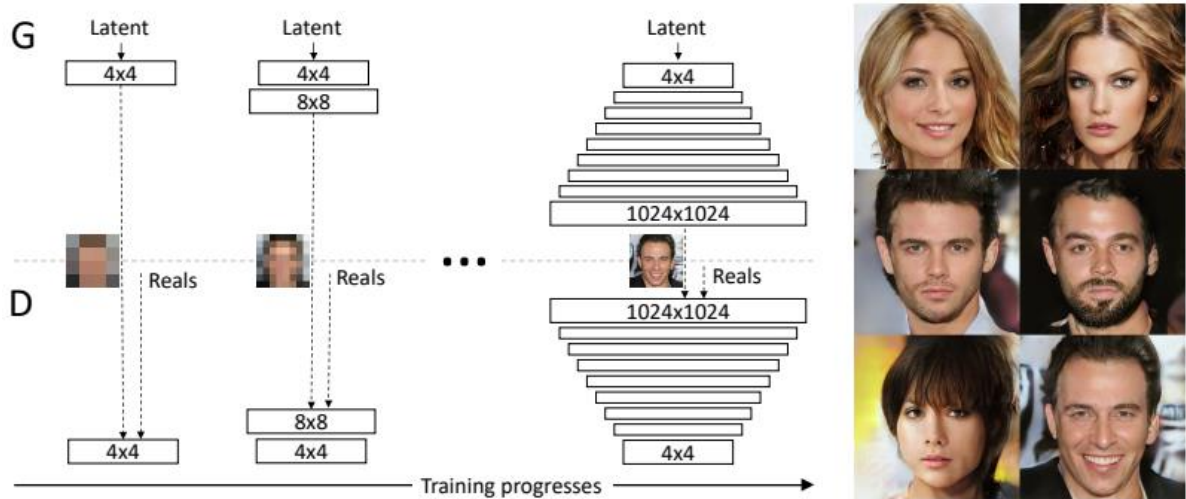


Рисунок 3.1 - Навчання починається як з генератора (G), так і з дискримінатора (D) з низьким просторовим дозволом 4×4 пікселів

По мірі прогресу навчання ми поступово додаємо шари до G і D, таким чином збільшуючи просторове дозвіл створених зображень. Усі існуючі шари залишаються відстежуваний протягом усього процесу. Тут $N \times N$ відноситься до згорткових шарів, що діють на $N \times N$ просторових дозвіл. Це дозволяє стабільний синтез у високій роздільній здатності, а також значно прискорює навчання. Праворуч ми показуємо шість прикладних зображень, згенерованих за допомогою прогресивного росту в 1024×1024 .

Ще одна перевага - скорочення часу на тренування. При прогресуючому зростанні генеративно-змагальній мереж більшість ітерацій виконуються з меншими роздільними здатностями, і порівнянна якість результату часто отримується в 2–6 разів швидше, залежно від кінцевої вихідної роздільної здатності. Ідея прогресивного розвитку генеративно-змагальних мереж поступово пов'язана з роботою Ван та ін. (2017), які використовують декілька дискримінаторів, які діють на різних просторових дозволах. Ця робота в свою чергу мотивована Дуругкар та ін. (2016), які використовують один генератор і декілька дискримінаторів одночасно, і Ghosh et al. (2017), які роблять навпаки з кількома генераторами та одним дискримінатором. Ієрархічна генеративно-змагальна мережа (Denton et al., 2015; Huang et al., 2016; Zhang et al., 2017) визначають генератор та дискримінатор для кожного рівня піраміди

зображень. Ці методи ґрунтуються на тому самому спостереженні, що і наша робота - що складне відображення від зображень із запізненням на зображення з високою роздільною здатністю легше засвоїти поетапно - але вирішальна різниця полягає в тому, що у нас є лише одна генеративно-змагальна мережа замість їх ієрархії. У контрасті для ранньої роботи над адаптаційно зростаючими мережами, наприклад, зростаючим нейронним газом (Fritzke, 1995) та нейро-еволюцією розширювальних топологій (Stanley & Miikkulainen, 2002), які жадібно ростуть мережі, ми просто відкладаємо введення попередньо налаштованих шарів. У цьому сенсі наш підхід нагадує пошарове навчання автокодер (Bengio et al., 2007).

3.2 Підвищення варіативності шляхом стандартного відхилення

Генеративно-змагальні мережі мають тенденцію охоплювати лише підмножину варіацій, знайдених у навчальних даних, і Саліманс та ін. (2016) пропонують як «міні-пакетну дискримінацію» як рішення. Вони не обчислюють статистику особливостей тільки з окремих зображень, але і через міні-серію, заохочуючи таким чином міні-партії генерували та тренували зображення для показу подібних статистичних даних. Це реалізується додаванням міні-партії шару до кінця дискримінатора, де шар дізнається великий тензор, який проектує активація вводу до масиву статистики. Окремий набір статистичних даних складається для кожного прикладу в a minibatch, і він приєднується до виходу шару, щоб дискримінатор міг використовувати статистику внутрішньо. Ми різко спрощуємо такий підхід, одночасно вдосконалюючи різні варіанти.

Наше спрощене рішення не має ні навчальних параметрів, ні нових гіперпараметрів. Ми спочатку обчислюємо стандартне відхилення для кожної функції у кожному просторовому розташуванні над міні-партією. Ми тоді середні ці оцінки за всіма можливостями та просторовими місцями отримуємо

єдине значення. Ми повторюємо цінують і з'єднують його з усіма просторовими місцями та над міні-партією, отримуючи одну додаткову (постійну) функцію карти. Цей шар можна було б вставити в будь-якому місці дискримінатора, але ми його знайшли найкраще вставити його до кінця (детальніше див. Додаток А.1). Ми експериментували з більш багатим набором статистики, але не змогли додатково покращити варіацію. Паралельно працюючи, Lin та співавт. (2017) надати теоретичну інформацію про переваги показу декількох зображень дискримінатору.

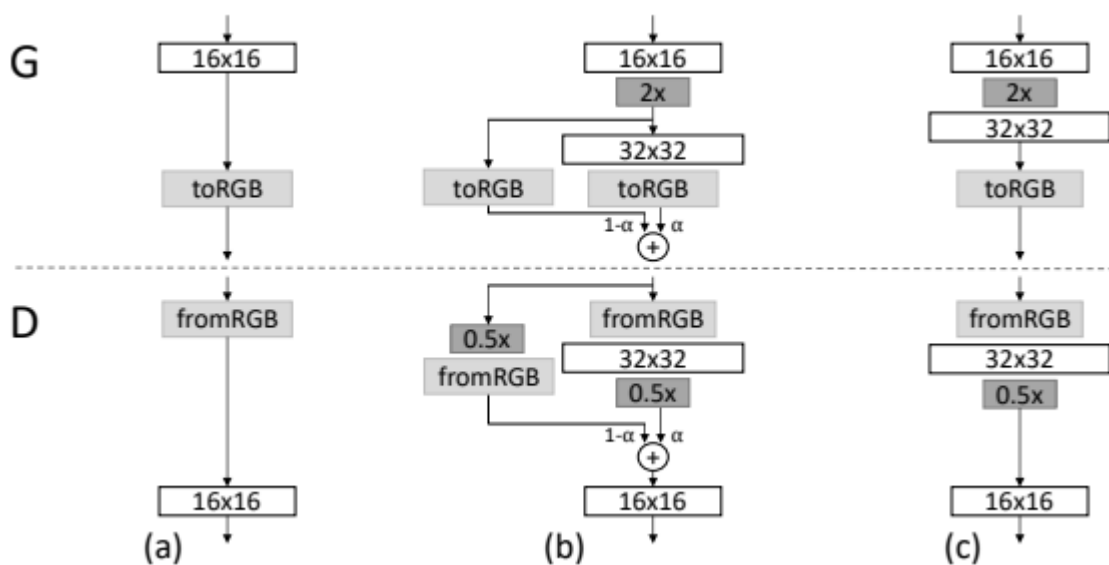


Рисунок 3.2 - При подвоєнні роздільної здатності генератора (G) та дискримінатора (D) ми згасаємо в нові шари плавно

Цей приклад ілюструє перехід від 16×16 зображень (a) до 32×32 образи (в). Під час переходу (b) ми розглядаємо шари, які працюють на більш високій роздільній здатності, як а залишковий блок, вага якого α лінійно збільшується від 0 до 1. Тут $2 \times 0,5 \times$ відносяться до подвоєння і вдвічі зменшити роздільну здатність зображення, використовуючи відповідно фільтрування найближчого сусіда та середнє об'єднання. ToRGB являє собою шар, який проектує особливості векторів на RGB кольори та fromRGB зворотний; обидва використовують 1×1 згортки. Навчаючи дискримінатора, ми подаємось у реальні образи які зменшуються в масштабі, щоб відповідати поточній

роздільній здатності мережі. Під час переходу до резолюції ми інтерполюємо між двома резолюціями реальних зображень, подібно до того, як генератор виводить поєднує дві резолюції.

Альтернативні варіанти варіаційної проблеми включають розкручування дискримінатора (Metz et al., 2016) регуляризувати його оновлення та "відштовхуючий регуляризатор" (Zhao et al., 2017), який додає новий термін втрати до генератора, намагаючись заохотити його ортогоналізувати функціональні вектори в міні-партії. Численні генератори Ghosh et al. (2017) також служать подібній меті. Ми визнаємо, що це рішення можуть збільшити варіацію навіть більше, ніж наше рішення - або, можливо, буде ортогональним для нього - але залиште детальне порівняння на більш пізній час.

3.3 Важливість індивідуальних внесків у умовах статистичної безпеки

Спочатку ми використаємо нарізану відстань Вассерстейна (SWD) та багато масштабну структурну схожість (MSSSIM) (Odena et al., 2017), щоб оцінити важливість нашого індивідуального внеску, а також сприйняти для себе оцінку самих показників. Ми зробимо це, будуючи на основі попередньої функції втрати стану-theart (WGAN-GP) та конфігурації тренувань (Gulrajani et al., 2017) без нагляду. Налаштування за допомогою наборів даних CELEBA (Liu et al., 2015) та LSUN BEDROOM (Yu et al., 2015) у 1282 році.

Нарізана відстань Wasserstein (SWD) між згенерованими та навчальними зображеннями та багато масштабною структурною схожістю (MS-SSIM) серед генерованих зображень для декількох навчальних установок на 128×128 . Для SWD кожен стовпець представляє один рівня Лаплайської піраміди, а остання дає в середньому чотири відстані.

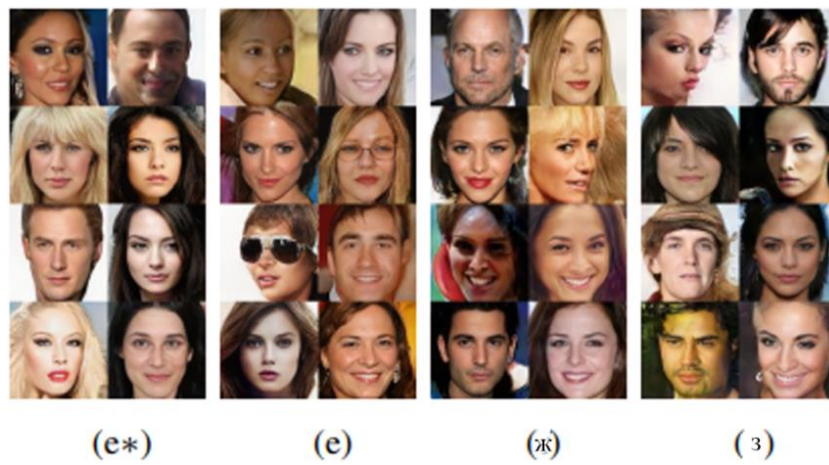
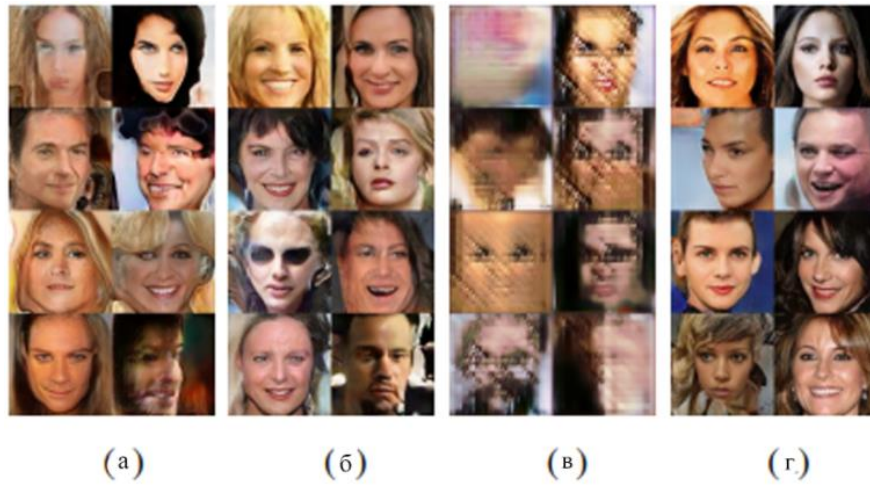


Рисунок 3.3 - (а) - (г) Приклади CELEBA



(к) Converged

Рисунок 3.4 – Кінцевий результат

Зауважте, що деякі зображення демонструють згладжування, а деякі – ні різкий - це недолік набору даних, який модель вчиться вірно копіювати.

CELEBA особливо добре підходить для такого порівняння, оскільки навчальні зображення містять помітні артефакти (згладжування, стиснення, розмиття), які важко відтворити генераторовірно. У цьому тесті ми посилюємо відмінності між конфігураціями тренувань, вибираючи відносно низьку мережеву структуру та припиняючи навчання, коли дискримінатору було показано загальну кількість 10-мільйонних реальних зображень. Тому такі результати не повністю збігаються.

У таблиці 3.1 перераховані числові значення для SWD та MS-SSIM в декількох навчальних конфігураціях, де наші індивідуальні внески кумулятивно включаються один за іншим у верхній частині базової лінії (Gulrajani та ін., 2017). Числа MS-SSIM були в середньому від 10000 пар згенерованих зображень, а SWD обчислювались. Створені зображення CELEBA з цих конфігурацій показані на рисунку 3.3. Зважаючи на обмеження місця, на рис. показано лише невелику кількість прикладів для кожного рядка таблиці, але значно ширший набір. Інтуїтивно, хороша метрика оцінювання повинна винагороджувати правдоподібні зображення, які демонструють велику кількість кольорів, текстур та точок зору. Однак MS-SSIM це не фіксує: ми можемо відразу побачити, що конфігурація (h) створює значно кращі зображення, ніж конфігурація (a), але MS-SSIM залишається приблизно незмінним, оскільки він вимірює лише варіацію між виходами, а не подібність до навчальний набір. З іншого боку, SWD вказує на явне поліпшення.

Перша конфігурація (a) тренінгу відповідає Gulrajani et al. (2017), що включає нормалізацію партії в генераторі, нормалізацію шару в дискримінаторі та розмір міні-партії 64. (б) дозволяє прогресивне зростання мереж, що призводить до більш чітких і правдоподібних вихідних зображень. SWD правильно вважає, що розподіл згенерованих зображень більше схожий на навчальний набір.

Наша основна мета - ввімкнути високу роздільну здатність, і для цього потрібно зменшити розмір мініатюр, щоб залишатися в межах доступного бюджету пам'яті. Ми ілюструємо впливаючі проблеми в (в) де ми зменшуємо розмір міні-партії з 64 до 16.

Створені зображення неприродні, що добре видно в обох показниках. У (г) ми стабілізуємо тренувальний процес, регулюючи гіперпараметрів, а також видаляючи пакетну нормалізацію та нормалізацію шару. Як проміжний тест (е *), ми дозволяємо міні-партію дискримінації (Salimans et al., 2016), яка дещо на диво не вдається покращити жодну з показників, включаючи MS-SSIM, яка вимірює варіації виходу.

На відміну від цього, наше стандартне відхилення (е) покращує середні бали та зображення SWD. Ми потім включити наші внески в (й) та (к), що призведе до загального поліпшення робочого періоду.

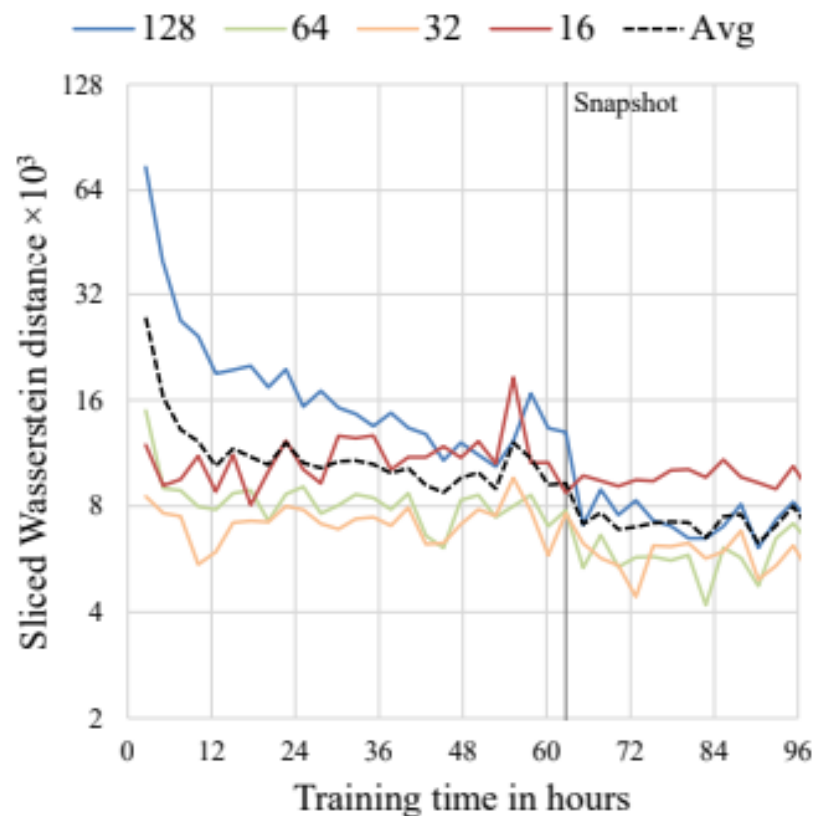


Рисунок 3.5 - Графік впливу відстані Вассерштейна у підході CELEBA

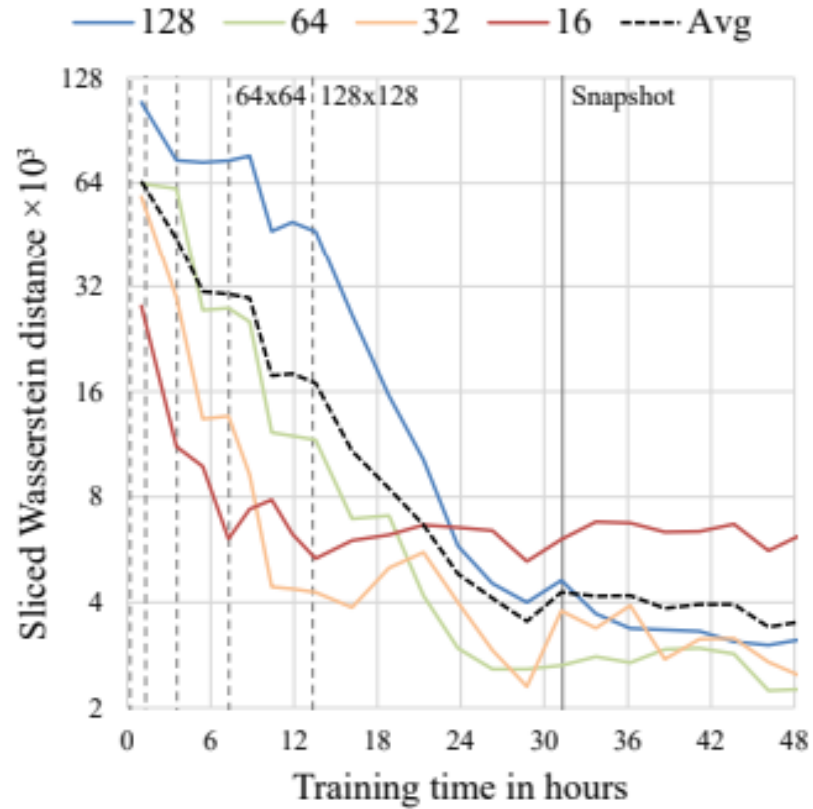


Рисунок 3.6 - Графік впливу відстані Вассерштейна у підході CELEBA з використанням GAN прогресивного зростання

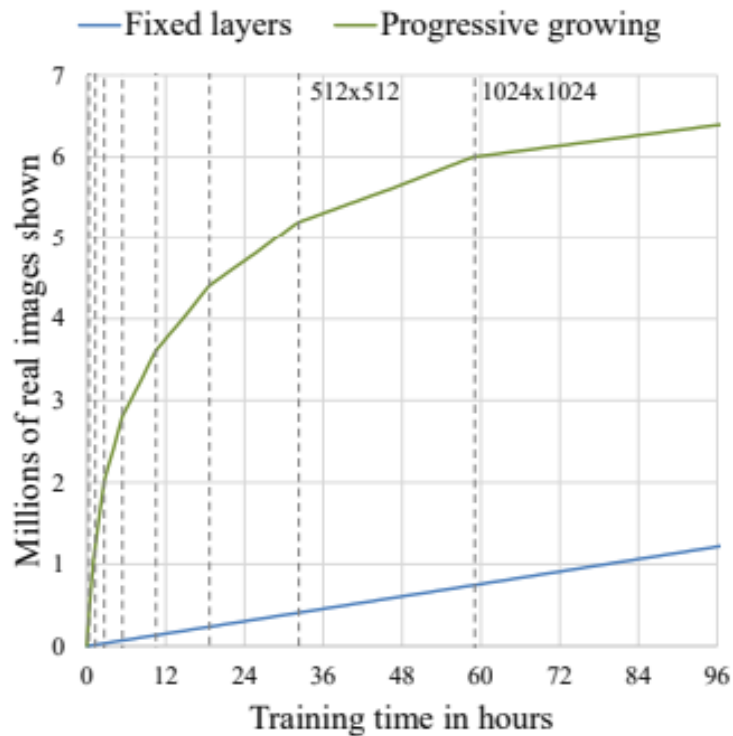


Рисунок 3.7 - Вплив прогресивного зростання на швидкість та конвергенцію тренувань.

Часи вимірювались для установки одного GPU за допомогою NVIDIA Tesla P100. (a) Статистична подібність стосовно час настінного годинника для Gulrajani та ін. (2017) за допомогою CELEBA у роздільній здатності 128×128 . Кожен графік являє собою нарізану відстань Вассерштейна на одному рівні Лаплакійської піраміди та вертикальній лінії вказує момент, коли ми припиняємо навчання в таблиці 3.1. (b) Той же графік з прогресивним зростанням увімкнено. Пунктирні вертикальні лінії позначають точки, в яких ми подвоюємо роздільну здатність G і D . (c) Вплив прогресивного зростання на швидкість сирого тренування в роздільній здатності 1024×1024 .

Нарешті, в (h) ми використовуємо не покалічену мережу і довше навчання – ми відчуваємо, що якість створених зображень принаймні порівнянна з найкращими опублікованими результатами досі.

3.4 Швидкість конвергенції та навчання

З точки зору метрики SWD та необробленого зображення пропускну здатність. Перші два сюжети відповідають навчальній конфігурації Gulrajani та ін. (2017) без і з прогресивним зростанням. Ми помічаємо, що прогресивний варіант пропонує дві основні переваги: він сходиться до значно кращого оптимального, а також скорочує загальний час тренувань приблизно коефіцієнт два. Поліпшення конвергенції пояснюється неявною формою навчання навчальних програм, що нав'язується поступово зростаючою мережевою ємністю. Без прогресивного зростання всі шари генератора та дискримінатора мають завдання одночасно знаходити короткі проміжні зображення як для масштабної варіації, так і для дрібномасштабної деталі. З прогресивним зростаючим, однак, існуючі шари з низькою роздільною здатністю, швидше за все, вже сходилися на початку, тож мережам покладається лише завдання уточнення уявлень шляхом все менших масштабних ефектів у міру впровадження нових шарів. Дійсно, ми бачимо на

рисунку 4 (б), що є найбільш масштабною статистикою Крива подібності (16) дуже швидко досягає свого оптимального значення і залишається послідовною протягом усього решта тренувань. Криві менших масштабів (32, 64, 128) вирівнюються одна за одну, як і роздільна здатність збільшився, але збіжність кожної кривої однаково послідовна. При непрогресивному навчанні. На рис. 3.4 (а) кожна шкала метрики SWD збігається приблизно в унісон, як можна було очікувати.

Швидкість від прогресивного зростання збільшується в міру зростання роздільної здатності. На рисунку 3.4 (в) показано хід навчання, вимірюваний кількістю реальних зображень, показаних дискримінатору, як функція час навчання, коли навчання прогресує аж до 10242 дозвіл. Ми це бачимо прогресивно. Зростання набирає значних результатів, оскільки мережі неглибокі та швидкі для оцінки початок. Після досягнення повної роздільної здатності пропускна здатність зображення дорівнює між двома методи. Сюжет показує, що прогресивний варіант досягає приблизно 6,4 мільйонів зображень у 96 годин, тоді як можна екстраполювати, що непрогресивний варіант займає близько 520 годин досягти тієї ж точки. У цьому випадку прогресивне зростання пропонує приблизно $5,4 \times$ прискорення.

ВИСНОВКИ

У рамках атестаційної роботи був розглянутий і реалізований метод генерації зображень на основі нейромереж.

Основна ідея - автоматизація виробництва контенту, який раніше вимагав зусиль дизайнерів.

Генеративні змагальні мережі - це багатообіцяючий клас генеративних моделей, який до цих пір стримувався нестабільною підготовкою та відсутністю належної метрики оцінювання. Генеративні змагальні мережі (GAN) досягли великого успіху завдяки своїй здатності генерувати реалістичні зразки. GAN складається з одного дискримінатора та одного генератора.

Дискримінатор намагається відрізнити реальні зразки від створених зразків, тоді як генератор підробляє реальні зразки, використовуючи інформацію від дискримінатора. GAN унікальна з багатьох інших генеративних моделей. Замість явної вибірки з розподілу ймовірностей GAN використовує глибоку нейронну мережу як прямий генератор, який генерує зразки з випадкових шумів.

Було продемонстровано здатність нашого методу до виродження модельних розподілів, ми розглянули проблему моделювання складного дискретного розподілу з GAN, генератор якого визначений у безперервному просторі.

Ми проаналізували конвергенцію навчання GAN з нової точки зору і розглянули, що колапс режиму відбувається через існування небажаних локальних рівноваг. Було проведено просте спостереження про те, як ситуація колапсу в режимі часто виявляє різкі градієнти функції дискримінатора навколо деяких реальних точок даних. Ця характеристика частково пояснює роботу раніше запропонованих WGAN та градієнтних штрафних санкцій та мотивує нашу нову штрафну схему.

Доведено, що GAN добре працює над кількома реалістичними завданнями, наприклад, зображення, малювання, розгортання та наслідування. Незважаючи на успіх у багатьох програмах, GAN дуже нестабільний у навчанні. Велика кількість класів об'єктів є особливо складним завданням для GAN через їх тенденцію до недооцінки ентропії в розподілі.

Ретельний підбір гіперпараметрів часто необхідний для зближення тренувального процесу. Часто вважають, що ця нестабільність викликана незбалансованою підготовкою дискримінаторів та генераторів. Оскільки дискримінатор використовує логістичні втрати, він швидко насичується, а його градієнт зникає, якщо створені зразки легко відокремити від реальних. Коли дискримінатор не надає градієнт, генератор припиняє оновлення.

Коли навчання GAN не є стабільним, генератор може генерувати зразки, подібні один одному. Ця відсутність різноманітності називається колапсом режиму, оскільки генератор може сприйматися як вибірки лише з деяких режимів розподілу даних.

GAN вимагають багато часу на тренування. На одному GPU тренування може займати години, а на одному CPU - більше одного дня. Незважаючи на складність настройки і, отже, використання, оглядам нових варіацій GAN і способам їх застосування присвячено багато статей.

Результати були апробовані у 32-ому збірнику Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського" та міжнародної науково-технічної конференції "Наукові теорії сьогодення та перспективи розвитку наукової думки" [44].

У якості подальшого напрямку дослідження має сенс розглянути питання нормалізації зображень за наявністю більш значних геометричних перетворень, а також збільшення кількості зображень для навчання генеративно-змагальної мережі зі 100 до 10000.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Habr.com. (2019). Фотографии из грубых набросков: как именно работает нейросеть NVIDIA GauGAN. [online] Available at: <https://habr.com/ru/company/itsumma/blog/447896/> [Accessed 5 Nov. 2019].
2. mattmazor.com. (2019). [online] Available at: <http://https://mattmazor.com/2015/03/17/a-step-by-step-backpropagation-example/> [Accessed 5 Nov. 2019].
3. Elib.spbstu.ru. (2019). [online] Available at: <http://elib.spbstu.ru/dl/2/v18-290.pdf/download/v18-290.pdf> [Accessed 5 Nov. 2019].
4. Salimans, T., Goodfellow, I., Zaremba, W., Radford, A. and Chen, X. (2019). Improved Techniques for Training GANs. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1606.03498v1.pdf> [Accessed 5 Oct. 2019].
5. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X. (2019). Improved Techniques for Training GANs. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1606.03498v1.pdf> [Accessed 5 Nov. 2019].
6. Habrahabr.ru. (2019). Пару слов о распознавании образов. [online] Available at: <https://habrahabr.ru/post/208090/> [Accessed 5 Nov. 2019].
7. Лепницкий, В. (2019). Генерация изображений и текстур. [online] Генерация изображений и текстур. Available at: <https://postnauka.ru/video/74128> [Accessed 5 Oct. 2019].
8. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – М.: ДМК Пресс, 2015. – 400 с.
9. En.wikipedia.org. (2019). Pattern recognition. [online] Available at: https://en.wikipedia.org/wiki/Pattern_recognition [Accessed 5 Nov. 2019].
10. Paperswithcode.com. (2019). Papers With Code : Image-to-Image Translation. [online] Available at: <https://paperswithcode.com/task/image-to-image-translation> [Accessed 5 Nov. 2019].

11. Phillipi.github.io. (2019). Image-to-Image Translation with Conditional Adversarial Networks. [online] Available at: <https://phillipi.github.io/pix2pix/> [Accessed 5 Oct. 2019].
12. Brownlee, J. (2019). How to Develop a Pix2Pix GAN for Image-to-Image Translation. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/how-to-develop-a-pix2pix-gan-for-image-to-image-translation/> [Accessed 5 Oct. 2019].
13. En.wikipedia.org. (2019). Computer vision. [online] Available at: https://en.wikipedia.org/wiki/Computer_vision [Accessed 5 Oct. 2019].
14. Путятин, Е. П. (1990). Обработка изображений в робототехнике. Москва.
15. Путятин, Є. П., Гороховатський, В. О., & Матаг, О. О. (2006). Методи та алгоритми комп'ютерного зору: навч. посіб. Харків: ТОВ «Компанія СМІТ».
16. Гороховатский, В. А. (2014). Структурный анализ и интеллектуальная обработка данных в компьютерном зрении.
17. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. International Journal of Science and Research (IJSR), 3(11), 2870-2877.
18. Gorokhovatskyi, V., Gorokhovatskyi, O., Yevgenyi, P., & Olena, P. (2018, August). Quantization of the Space of Structural Image Features as a Way to Increase Recognition Performance. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (pp. 464-467). IEEE.
19. Putyatin, E.P. (2019, September 12). Normalization and image recognition. Retrieved from <http://sumschool.sumdu.edu.ua/is-02/ms/lectures/pytyatin/pytyatin.htm>.
20. En.wikipedia.org. (2019). Generative adversarial network. [online] Available at: https://en.wikipedia.org/wiki/Generative_adversarial_network [Accessed 5 Nov. 2019].

21. Ansah H. (2019). Unpaired Image to Image Translation with CycleGAN. [online] Paperspace Blog. Available at: <https://blog.paperspace.com/unpaired-image-to-image-translation-with-cyclegan/> [Accessed 5 Nov. 2019].
22. mdpi.com. (2019). Image-to-Image Translation Using Identical-Pair Adversarial Networks. [online] Available at: <https://www.mdpi.com/2076-3417/9/13/2668/htm> [Accessed 5 Oct. 2019].
23. Isola, P., Zhu, J., Zhou, T. and A. Efros, A. (2019). Image-to-Image Translation with Conditional Adversarial Networks. [online] researchgate.net. Available at: https://www.researchgate.net/publication/320966887_Image-to-Image_Translation_with_Conditional_Adversarial_Networks [Accessed 5 Nov. 2019].
24. Paperswithcode.com. (2016). State-of-the-art table for Image-to-Image Translation on Cityscapes Labels-to-Photo. [online] Available at: <https://paperswithcode.com/sota/image-to-image-translation-on-cityscapes> [Accessed 5 Nov. 2019].
25. Путятин Е.П., Аверин С.И. Обработка изображений в робототехнике. М: Машиностроение, 1990. 320 с.
26. Гиренко, А. В., Ляшенко, В. В., Машталир, В. П., & Путятин, Е. П. (1996). Методы корреляционного обнаружения объектов. Харьков: АО «БизнесИнформ, 112.
27. Путятин, Е. П. (2012). Нормализация и распознавание изображений [Электронный ресурс]/Сумский гос. ун-т, летняя научно-практическая школа «Интеллектуальные системы». Режим доступа: URL: <http://sumschool.sumdu.edu.ua/is-02/rus/lectures/pytyatin/pytyatin.htm>.
28. Панченко, Д. С., & Путятин, Е. П. (1999). Сравнительный анализ методов сегментации изображений. Радиоэлектроника и информатика, (4 (9)).
29. Путятин, Е. П., & Гороховатский, А. В. (2006). Построение инвариантных моментных признаков изображений с использованием одномерных проекций.

30. Гороховатский, А. В., & Путятин, Е. П. (2005). Применение преобразования Радона для нормализации изображений.
31. Путятин, Е. П., Яковлева, Е. В., & Любченко, В. А. (1998). Разложение матрицы центроаффинного преобразования для нормализации изображений. Радиоэлектроника и информатика, (4 (5)).
32. Gorokhovatskyi, V., Gadetska, S., & Ponomarenko, R. (2019, May). Recognition of Visual Objects Based on Statistical Distributions for Blocks of Structural Description of Image. In International Scientific Conference “Intellectual Systems of Decision Making and Problem of Computational Intelligence” (pp. 501-512). Springer, Cham.
33. Гороховатский, В. А., Кацалап, С. Ф., & Путятин, Е. П. (1986). Анализ изображений в условиях локальных искажений. Автометрия, (6), 46.
34. Gorokhovatskyi, V., Vasylchenko, A., Manko, K., & Ponomarenko, R. (2018). Дослідження модифікацій методу встановлення релевантності зображень об'єктів за описами у вигляді множини дескрипторів ключових точок. Системи управління, навігації та зв'язку. Збірник наукових праць, 5(51), 74-78.
35. Любченко, В. А., & Путятин, Е. П. (2000). Математические модели нормализации перспективы. Проблемы бионики, (53), 22-27.
36. Гороховатский, А. В., & Путятин, Е. П. (2007, September). Распознавание изображений на основе признаков проекций. In Искусственный интеллект. Интеллектуальные и многопроцессорные системы: Матер. 8-й Межд. научно-техн. конф.–Донецк: ИПИИ (pp. 203-207).
37. Kinoshenko, D., Mashtalir, V., & Shlyakhov, V. (2007). A partition metric for clustering features analysis.
38. Mashtalir, V., Mikhnova, E., Shlyakhov, V., & Yegorova, E. (2006, November). A novel metric on partitions for image segmentation. In 2006 IEEE International Conference on Video and Signal Based Surveillance (pp. 18-18). IEEE.
39. Kinoshenko, D., Mashtalir, V., Yegorova, E., & Vinarsky, V. (2005, July). Hierarchical partitions for content image retrieval from large-scale database. In

International Workshop on Machine Learning and Data Mining in Pattern Recognition (pp. 445-455). Springer, Berlin, Heidelberg.

40. Kagramanyan, A., Mashtalir, V., & Shlyakhov, V. (2008). Multialgebraic systems in information granulation.

41. Chupikov, A., Kinoshenko, D., Mashtalir, V., & Shcherbinin, K. (2006, July). Image retrieval with segmentation-based query. In International Workshop on Adaptive Multimedia Retrieval (pp. 207-221). Springer, Berlin, Heidelberg.

42. Путятин, Е. П., & Сердюченко, В. Я. (1972). Математические модели явления краевого контраста ахроматического зрения человека (обзор). Проблемы бионики, (9), 219.

43. Путятин, Е. П., Гороховатский, В. А., Добрынин, А. А., Ересько, Ю. Н., & Сытник, О. В. (1992). Определение параметров объектов по серии изображений. Автометрия, (4), 18-23.

44. Вовк А. С., Дослідження реконструкції моделей в просторі на основі зображень [Текст] / А. С. Вовк, наук. керівник Є.П. Путятін // Наукові теорії сьогодення та перспективи розвитку наукової думки: матеріали міжнародної наукової конференції (Т. 1), 11 жовтня, 2019 рік. Київ, Україна: МЦНД.