ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методу розпізнавання рукописних зображень на основі варіаційного автокодувальника»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Олександр КОТУЛ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-61
Олександр КОТУЛ

Керівник: Олена НЕГОДЕНКО
к.т.н., доцент

Рецензент: _____
науковий ступінь, вчене звання *Ім'я, ПРІЗВИЩЕ*

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____ Ірина ЗАМРІЙ
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Котулу Олександру Юрійовичу

1. Тема кваліфікаційної роботи: «Розробка методу розпізнавання рукописних зображень на основі варіаційного автокодувальника»

керівник кваліфікаційної роботи Олена НЕГОДЕНКО к.т.н., доцент,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023 р. №145.

2. Строк подання кваліфікаційної роботи «29» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, матеріали переддипломної практики, методи розпізнавання та генерації зображень.

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

1.Огляд предметної області.
2.Аналіз методів розпізнавання зображень.
3.Огляд методів для генерації зображень.
4.Проектування методу та аналіз отриманих результатів

5. Перелік графічного матеріалу: *презентація*

1. Методи розпізнавання зображень
2. Характеристики методів розпізнавання
3. Методи генерації зображень
4. Модель розпізнавання зображень на основі варіаційного автокодувальника.
5. Аналіз отриманих результатів.

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Аналіз існуючих методів розпізнавання зображень	06.11-12.11.23	
3	Моделювання алгоритму для підвищення ефективності	13.11-22.11.23	
4	Розробка моделей та методів	23.11-30.11.23	
5	Аналіз отриманих результатів	30.11-10.12.23	
7	Оформлення роботи: вступ,висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти _____ Олександр КОТУЛ
(підпис)

Керівник
кваліфікаційної роботи _____ Олена НЕГОДЕНКО
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 71 стор., 2 табл., 27 рис., 22 джерел.

Мета роботи – покращення точності розпізнавання рукописних зображень за рахунок варіаційного автокодувальника.

Об'єкт дослідження – процес розпізнавання рукописних зображень.

Предмет дослідження – метод для розпізнавання зображень на основі варіаційного автокодувальника.

Короткий зміст роботи:

У роботі проведено дослідження методів розпізнавання зображень, таких як CNN, K-NN, SVM. Проаналізовано переваги та недоліки кожного методу, та визначено показники які досягають ці моделі на малих навчальних вибірках. Також була виявлена одна з основних проблем при використанні цих методів.

У сучасному машинному навчанні, де доступ до великих обсягів реальних даних часто обмежений, проблема відсутності навчальних прикладів постає як ключова. Розглядаються та аналізуються генеративні методи , які дозволяють збільшити обсяг даних шляхом введення різноманітних перетворень та модифікацій вихідного набору.

Особлива увага приділяється генеративним моделям, таким як варіаційний автокодувальник (VAE). Детально розглядається їх здатність синтезувати нові, реалістичні образи та їх вплив на підвищення ефективності моделей. Розглядається здатність створювати реалістичні зображення, підкреслюється важливість використання таких методів у сценаріях з невеликою кількістю доступних даних.

КЛЮЧОВІ СЛОВА: РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, СИНТЕЗ ДАНИХ.

ABSTRACT

Text part of the master's qualification work: 71 pages, 27 pictures, 2 table, 22 sources.

The purpose of the work – improving the accuracy of recognition of handwritten images due to the variational autoencoder.

Object of research – the process of recognizing handwritten images.

Subject of research – method for image recognition based on variational autoencoder.

Summary of the work: The research of image recognition methods such as CNN, K-NN, SVM is carried out in the work. The advantages and disadvantages of each method are analyzed, and the indicators achieved by these models on small training samples are determined. One of the main problems when using these methods was also revealed.

In modern machine learning, where access to large volumes of real data is often limited, the problem of lack of training examples appears as a key one. Generative methods are considered and analyzed, which allow to increase the amount of data by introducing various transformations and modifications of the original set.

Particular attention is paid to generative models such as the variational autoencoder (VAE). Their ability to synthesize new, realistic images and their influence on increasing the efficiency of models is considered in detail. The ability to create realistic images is examined, emphasizing the importance of using such techniques in scenarios with little available data.

KEYWORDS: IMAGE RECOGNITION, NEURAL NETWORKS, MACHINE LEARNING, DATA SYNTHESIS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CNN – convolutional neural network

VAE – Variational autoencoder

SVM – support vector machines

GAN – Generative adversarial network

K-NN – k-nearest neighbors

AR – Autoregressive model

ШІ – штучний інтелект

ШНМ – штучних нейронних мереж

ВСТУП

Розвиток сучасних технологій в сфері штучного інтелекту, машинного навчання та комп'ютерного зору відкриває безліч нових можливостей для реалізації інноваційних систем розпізнавання та обробки зображень. Однією з важливих галузей застосування цих технологій є розпізнавання рукописних зображень, що знаходить широке застосування в автоматизованих системах, таких як розпізнавання символів, підписів, а також в аналізі медичних зображень та багатьох інших областях.

Варіаційні автокодувальники (VAE) представляють собою потужний клас моделей глибокого навчання, здатних до ефективного вивчення унікальних репрезентацій даних та генерації нових, подібних до них. Використання VAE для завдань розпізнавання рукописних зображень може значно покращити якість та ефективність систем автоматичного аналізу та інтерпретації графічних даних.

Метою даної дипломної роботи є розробка нового методу розпізнавання рукописних зображень на основі варіаційного автокодувальника. Цей метод буде орієнтований на вдосконалення точності розпізнавання, а також на здатність генерації нових зображень.

Дана робота включає в себе ретельний аналіз існуючих підходів до розпізнавання рукописних зображень, детальний огляд теорії варіаційних автокодувальників, а також розробку та експериментальну валідацію запропонованого методу. Очікується, що отримані результати не тільки підтвердять ефективність розробленого методу, але і відкриють нові можливості для вдосконалення систем розпізнавання та генерації рукописних зображень в майбутньому.

Таким чином, дипломна робота має на меті внести свій внесок у розвиток області комп'ютерного зору та машинного навчання, зосереджуючись на покращенні розпізнавання рукописних зображень з використанням інноваційних методів на основі варіаційних автокодувальників.

Мета: покращення точності розпізнавання рукописних зображень за рахунок варіаційного автокодувальника.

Об’єкт: процес розпізнавання рукописних зображень.

Предмет: метод для розпізнавання зображень на основі варіаційного автокодувальника.

Наукова новизна: результати дослідження пропонують метод розпізнавання рукописних зображень, який здатен працювати з малим набором навчальних даних. Основою методу є варіаційний автокодувальник, завданням якого є аугментація навчальної вибірки.

Практична значущість отриманих результатів: полягає в тому, що розроблений метод може показати кращі результати в розпізнаванні зображень, в порівнянні з звичайними методами

Завдання дослідження:

1.Аналіз методів розпізнавання розбражень

2. Дослідження проблеми використання згорткових мереж, з малою навчальною вибіркою.

3.Аналіз методів для генерації зображень.

4. Дослідження моделі варіаційного автокодувальника.

5. Генерація зображень подібних до навчальної вибірки.

6.Підвищення ефективності навчання згорткової мережі, за рахунок навчання на розширеній вибірці з згенерованими зображеннями

Апробація результатів та публікації:

Тези доповідей:

Котул О.Ю. Використання нейромереж для генерації зображень // Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і Світу». – Київ: ДУІКТ, 2023. с.185-186

Котул О.Ю. Аналіз проблем навчання нейронних мереж // XVII Міжнародна науково-практична конференція «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» – Дніпро: УДУНТ, 2023 с.117

Стаття:

Котул О.Ю. Аналіз методів для розпізнавання та синтез зображень для розширення навчальної вибірки // Зв’язок 2023 (прийнято до друку)

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Визначення предметної області

У світлі стрімкого розвитку технологій машинного навчання та штучного інтелекту, питання розпізнавання рукописних зображень стає надзвичайно актуальним і важливим. Ця проблема виникає у контексті широкого спектру застосувань, таких як розпізнавання символів, підписів, аналіз медичних зображень, автоматизація документообігу та інші галузі, де необхідна точна та ефективна обробка графічних даних. Розпізнавання рукописних зображень відкриває можливості для створення інтелектуальних систем, здатних автоматично аналізувати та інтерпретувати інформацію, що подана у вигляді рукописних символів. Проте, існуючі методи розпізнавання мають свої обмеження такі як час навчання, потреба в обчислювальних ресурсах, кількість навчальних даних, і виникає потреба у нових та більш ефективних підходах для вирішення цієї проблеми.

1.2 Актуальність теми

За останні роки все більше набувають популярності такі напрямки як нейронні мережі та машинне навчання. Вони можуть бути використані для різних завдань, таких як класифікація, регресія, визначення об'єктів у зображеннях, розпізнавання мови, розробка прогнозів або прийняття рішень на основі даних. Але для ефективного та якісного навчання нейромережі необхідно якомога більше навчальних даних. В залежності від завдання це може бути різний матеріал, для прикладу нейромережу яка розпізнає зображення, необхідно навчати саме на таких зображеннях[20]. Але може виникнути ситуація, коли для якісного навчання нейромережі цих зображень недостатньо, в такому випадку

головним завданням є аугментація навчальної вибірки за допомогою генерації зображень.

На сьогоднішній день для розпізнавання зображень існують різні методи. Для прикладу методи засновані на ознаках (SIFT, HOG)[14]. В роботі [18] зазначається що метод ці методи мають гіршу ефективність ніж глибокі нейронні мережі. Не поганий результат також відображають методи на основі опорних векторів - це методи, які використовують опорні вектори для класифікації зображень [6].

Опорні вектори - це зображення, які є найбільш відокремленими один від одного в евклідовому просторі. Було розроблено кілька систем класифікації, заснованих на SVM, для розпізнаванням рукописних цифр, і деякі обнадійливі результати були зафіксовані, наприклад, в [9, 10, 11].

Одним із методів для розпізнавання зображень є метод k-найближчих сусідів [8]. Автори розглядають переваги та недоліки методу k-NN. Переваги методу включають його простоту реалізації, ефективність та здатність працювати з високомірними даними. Недоліки методу включають його чутливість до шуму в даних та його можливість перенавчання. В даній роботі також розглядаються різні методи підвищення ефективності методу k-NN. Зазначається, що метод k-NN є все ще актуальним методом розпізнавання зображень, незважаючи на появу більш складних методів. Це пояснюється простотою реалізації, ефективністю та широким спектром застосувань методу k-NN. Такий підхід, як повідомляється, дає досить хороші результати, наприклад, [12, 13], хоча слід зазначити, що він вимагає значних обчислювальних потужностей в процесі класифікації.

В роботі [7] було проведене порівняння алгоритму логістичної регресії та штучних нейронних мереж (ШНМ) для задачі розпізнавання рукописних символів. Логістична регресія - це статистичний метод, який використовується для прогнозування ймовірності того, що певна подія відбудеться. Він використовується в різних задачах машинного навчання, таких як класифікація, прогнозування та оцінка ризику. В обох підходах важливу роль грає машинне навчання. В результаті дослідження встановлено що використання ШНМ є ефективнішим (за точністю

розпізнавання) для задачі розпізнавання символів. Точність розпізнавання для ШНМ склала 97,52%, а для алгоритму логістичної регресії 95.08 %. Слід зазначити що тренування відбувалося на наборі 5000 зображень.

Глибоке навчання, зокрема згорткові нейронні мережі (CNN), також можуть ефективно впоратися із завданнями класифікації та визначення об'єктів на зображеннях[1, 2]. Згорткові нейронні мережі (CNN) вважаються найбільш популярними в застосуванні для класифікації об’єктів на зображеннях[3].

В роботі [4] було проведено дослідження та порівняння різних архітектур згоркових мереж для розпізнавання символів, зокрема використання передньо навчених моделей, таких як ResNet, Inception, та MobileNet в результаті було визначено, що обсяг навчальної вибірки сильно впливає на надійність розпізнавання символів.

Для збільшення обсягу навчальної вибірки можна використати Image Data Generator з пакету tensorflow, який випадково обертає, маштабує, стискає чи витягує зображення[15]. Також цікавим рішенням може бути використання генеративних нейромереж, які можуть копіювати стилі чи ознаки зображень з навчальної вибірки для створення унікальних зображень.

В роботах [5,16,17] було представлено найпопулярніші типи генеративних мереж: варіаційний автоенкодер (VAE), генеративні змагальні мережі(GAN), авторегресивні мережі(AR). В результаті виявлено наступне: VAE може бути хорошим вибором, якщо важлива можливість варіювання та робота з латентним простором; GAN може бути оптимальним вибором, якщо найважливішою є якість та реалістичність генерованих зображень [19]; авторегресивні мережі можуть бути ефективними, якщо важлива генерація зображень з урахуванням структури та локальних залежностей.

Застосування варіаційних автокодувальників для генерації рукописних зображень виступає як інноваційний підхід, який може розширити можливості генерації нових даних та поліпшити якість та точність розпізнавання. Дослідження в цьому напрямку може привести до створення більш ефективних та виразних

систем розпізнавання рукописних зображень, що знайдуть застосування у різних галузях життя та бізнесу.

1.3 Розпізнавання зображень: поняття та особливості

Розпізнавання зображень (computer vision) — це галузь машинного навчання, яка займається автоматичною обробкою зображень з метою виявлення, класифікації та розуміння об'єктів та явищ, що зображені на них. Розпізнавання зображень має широке застосування в різних сферах, включаючи комп'ютерну візуалізацію, охорону здоров'я, безпеку та автоматизацію.

1.3.1 Основні поняття:

Піксель (Pixel) - найменша одиниця растрового (bitmap) зображення. Кожен піксель представляється числовим значенням, яке вказує його колір, інтенсивність чи інші характеристики.

Об'єкт (Object) - це конкретний об'єкт або елемент, який система спробує ідентифікувати на зображенні.

Класифікація (Classification) - процес призначення об'єкта до конкретного класу або категорії. У розпізнаванні зображень це означає визначення, до якої категорії належить об'єкт на зображенні.

Детекція (Detection) - виявлення та локалізація об'єктів на зображенні. Це включає визначення координат та межі об'єктів.

Сегментація (Segmentation) - розділення зображення на окремі сегменти або області, що відповідають різним об'єктам чи частинам зображення.

Навчання з учителем - багато сучасних систем розпізнавання зображень базуються на навчанні з учителем, де модель навчається на позначених даних, що містять зображення та відповідні мітки класів.

Глибоке навчання (Deep Learning) - використання глибоких нейронних мереж для автоматичного визначення признаків та шарів абстракції для розпізнавання складних зображень.

Перенос вивчення (Transfer Learning) - застосування знань, набутих на одному наборі даних, до розв'язання задачі розпізнавання на іншому наборі даних.

Аугментація даних (Data Augmentation) - застосування технік для збільшення різноманітності тренувального набору шляхом зміни зображень.

1.3.2 Особливості розпізнавання зображень:

Розпізнавання зображень є складним завданням, оскільки зображення є високовимірними об'єктами. Це означає, що зображення містять багато інформації, яку необхідно обробити. Крім того, зображення можуть бути забруднені шумом або мати інші дефекти, які можуть ускладнити розпізнавання.

Для вирішення цих проблем у розпізнаванні зображень використовуються різні методи, такі як:

- Фільтрація — видалення шуму та інших артефактів зображення.
- Добування ознак — вибір характеристик зображення, які є важливими для розпізнавання. Цей процес є критичним для успішного розпізнавання зображень, оскільки від якості виділених ознак залежить точність класифікатора.
- Вирівнювання зображення — перетворення зображення так, щоб його характеристики були більш однорідними. Цей процес може бути корисним для розпізнавання зображень, які були отримані в різних умовах освітлення або з різних ракурсів.
- Штучні нейронні мережі — методи машинного навчання, які можуть бути використані для розпізнавання зображень з високою точністю.

- Локалізація та виділення (Localization and Highlighting) - визначення та виділення конкретних областей або об'єктів на зображенні.

1.3.3 Основні етапи розпізнавання зображень

Розпізнавання зображень можна описати як процес, що складається з наступних етапів:

1. Вхід— зображення, яке необхідно розпізнати.

2. Обробка — перетворення зображення в формат, який може бути оброблений комп'ютером. Цей етап може включати такі операції, як освітлення, контрастування, згладжування, видалення шуму та ін.

3. Виділення ознак — вибір характеристик зображення, які є важливими для розпізнавання. Цей етап може включати такі операції, як визначення контурів, вимірювання розмірів та форм, виявлення текстури та ін.

4. Класифікація — визначення класу, до якого належить зображення. Цей етап може бути реалізований за допомогою різних методів машинного навчання, включаючи наївний байесовський класифікатор, логістичну регресію, штучні нейронні мережі та ін.

1.3.4 Задачі розпізнавання зображень

Розпізнавання зображень може бути використано для вирішення різних завдань, включаючи:

- Розпізнавання об'єктів — визначення типу об'єкта, який зображений на зображенні. Наприклад, розпізнавання лиць, розпізнавання рукописних цифр, розпізнавання дорожніх знаків тощо.
- Розпізнавання сцен — визначення вмісту сцени, яка зображена на зображенні. Наприклад, розпізнавання будівель, розпізнавання людей, розпізнавання дій тощо.

- Розпізнавання тексту — визначення тексту, який зображений на зображенні. Наприклад, розпізнавання рукописного тексту, розпізнавання друкованого тексту тощо.
- Розпізнавання зображень є активною областю досліджень. У останні роки спостерігається значний прогрес у цій галузі, що пов'язано з розвитком нових методів машинного навчання, таких як штучні нейронні мережі.
- Завдяки цьому прогресу розпізнавання зображень стало більш точним і ефективним. Це призвело до широкого поширення цієї технології в різних областях.

1.4 Огляд існуючих методів розпізнавання рукописних зображень

1.4.1 Метод k-найближчих сусідів

Метод k-найближчих сусідів (k-Nearest Neighbors або k-NN) є простим та ефективним алгоритмом для класифікації та регресії, особливо в ситуаціях, де важко або неможливо побудувати аналітичні моделі. Основна ідея полягає в тому, щоб призначити клас (або передбачити значення) нового прикладу, засновуючись на класах його k найближчих сусідів у просторі ознак.

Переваги k-найближчих сусідів:

- Простота та інтуїтивність k-NN - дуже простий алгоритм, легкий для реалізації та зрозуміння. Він не вимагає складних моделей чи глибокого розуміння даних.
- Відсутність уявлення про розподіл - k-NN не передбачає або не робить припущень про розподіл даних, що робить його відмінним для завдань, де розподіл може бути нелінійним чи складним.
- Здатність до вирішення нелінійних завдань - він може добре працювати для класифікації випадків, коли рішення не може бути виражено лінійними або аналітичними методами.

- Ефективність при невеликих розмірах даних - коли обсяг даних невеликий, k-NN може бути ефективним, особливо якщо вимагається висока точність прогнозу.

Недоліки k-найближчих сусідів:

- Високі обчислювальні витрати - обчислювальні витрати зростають з розміром тренувального набору, оскільки для кожного нового прикладу потрібно розраховувати відстані до всіх точок у тренувальному наборі.
- Чутливість до шуму та викидів - k-NN може бути чутливим до шуму та викидів у даних, оскільки він розглядає всі приклади однаково, незалежно від їх значимості чи надійності.
- Неефективність в високорозмірних просторах - у високорозмірних просторах всі точки стають однаково віддаленими, що може знизити ефективність методу.
- Необхідність вибору параметра k - вибір вірної кількості сусідів (k) може впливати на результати, і важко визначити оптимальне значення, яке підходить для всіх ситуацій.
- Залежність від відстані - результати можуть сильно змінюватися в залежності від вибраної метрики відстані, і вибір відповідної метрики є важливим аспектом.

Хоча k-NN може бути ефективним для деяких завдань, особливо в простих сценаріях, його використання може бути обмеженим для складних та великих наборів даних.

1.4.2 Метод опорних векторів

Одним із методів для розпізнавання зображень, є метод опорних векторів (Support Vector Machine, SVM). Він являється алгоритмом машинного навчання, який використовується для завдань класифікації та регресії. Основні концепції, які визначають SVM:

- Гіперплощина (Hyperplane), (n-1)-мірна площина в n-вимірному просторі ознак, яка розділяє дані різних класів. Для бінарної класифікації це гіперплощина повинна бути максимально віддаленою від точок кожного класу.
- Опорні вектори (Support Vectors), точки даних, які лежать найближче до гіперплощини і визначають її положення та орієнтацію. Вони грають ключову роль у визначенні гіперплощини та розрахунку відстані між класами.
- Margin - відстань між гіперплощиною та найближчими точками кожного класу (опорними векторами). Основна ідея полягає в тому, щоб максимізувати цей margin для забезпечення кращої генералізації на нових даних.
- Ядро - функція, яка дозволяє вирішити нелінійні проблеми, розширюючи простір ознак. SVM використовується для роботи з даними, які не можна ефективно розділити лінійно в оригінальному просторі ознак.
- Параметр регуляризації (C) - визначає компроміс між максимізацією margin і мінімізацією помилок класифікації. Велике значення C призводить до меншого margin, але точкова класифікація стає більш точною.

Основною метою SVM є знаходження гіперплощини в просторі ознак, яка найкращим чином розділяє точки даних різних класів. У випадку бінарної класифікації, ця гіперплощина повинна максимізувати відстань між найближчими точками кожного класу, які називаються опорними векторами. Даний метод має свої переваги та недоліки

Переваги:

- ефективно працює в просторах високої розмірності, що робить його відмінним вибором для задач з багатьма ознаками чи великою кількістю ознак.

- SVM забезпечує математично обґрунтований метод для пошуку оптимальної гіперплощини, що розділяє класи. Це може забезпечити впевненість в правильності рішення.
- Зазвичай ефективний, коли об'єм даних обмежений, і немає необхідності у великій кількості навчальних прикладів.
- За допомогою параметра регуляризації C можна контролювати помилки класифікації на тренувальному наборі. Це дозволяє адаптувати модель до рівня помилок, прийнятного для конкретної задачі

Недоліки:

- чутливий до масштабу ознак, тому потрібна нормалізація або стандартизація даних для досягнення оптимальних результатів.
- Визначення ядра та налаштування параметрів (наприклад, параметр регуляризації C) може бути нетривіальним завданням.
- Тренування моделі SVM може бути дуже обчислювально витратним для великих обсягів даних та великої кількості ознак.
- Основний алгоритм SVM розроблений для бінарної класифікації, і для багатокласових задач використовуються різні стратегії (наприклад, один-проти-одного або один-проти-всі).
- Схильність до перенавчання, особливо коли дані несбалансовані, тобто кількість екземплярів в різних класах різниться.

Порівнюючи з конволюційними нейронними мережами (CNN), SVM зазвичай використовуються для класифікації структурованих даних, в той час як CNN ефективніше для завдань розпізнавання образів, зокрема обробки зображень.

1.4.3 Згорткові нейронні мережі

Згорнуткові нейронні мережі (Convolutional Neural Networks, CNNs або ConvNets) - це тип глибоких нейронних мереж, спеціально розроблений для обробки та аналізу зображень. CNN широко використовуються в задачах комп'ютерного зору, розпізнавання об'єктів, відображення зображень та багатьох

інших областях, де важлива обробка візуальної інформації. Основною відмінністю CNN від інших типів нейронних мереж є використання згорткових шарів для ефективної роботи з зображеннями. Основні компоненти CNN включають:

- Згорткові шари (Convolutional Layers) - використовуються для виявлення локальних ознак та особливостей в зображеннях. Кожен шар має набір фільтрів, які згортаються зі вхідними даними для створення фільтрованих зображень.
- Шари підвибірки (Pooling Layers) - використовуються для зменшення просторового розміру фільтрованих зображень, зберігаючи важливі особливості. Популярний тип підвибірки - максимальна підвибірка, де обирається максимальне значення у певному регіоні.
- Повністю з'єднані шари (Fully Connected Layers) - використовуються для згортки інформації та прийняття рішення на основі виявлених особливостей. Ці шари можуть включувати класифікаційні шари для визначення категорії об'єкта на зображенні.
- Функції активації - такі як ReLU (Rectified Linear Unit), часто використовуються для введення нелінійності в модель та покращення її здатностей до вивчення складних залежностей.

Основні переваги CNN:

- Здатність до ієрархічного вивчення особливостей - згорткові шари вивчають локальні шаблони, а повністю з'єднані шари об'єднують ці шаблони для розпізнавання вищих рівнів особливостей та структур.
- Параметрична ефективність - використання параметрів для обмежених регіонів зображення робить CNN параметрично ефективним в порівнянні з повністю з'єднаними мережами.
- Інваріантність до зсувів та масштабування - згорткові шари володіють інваріантністю до зсувів та масштабувань, що робить CNN ефективним для розпізнавання об'єктів у різних масштабах та положеннях на зображенні.

Недоліки:

- Потреба в великих обсягах даних - CNN часто потребує великої кількості даних для ефективного навчання, щоб уникнути перенавчання.
- Витрати обчислювальних ресурсів - згорткові шари можуть бути обчислювально витратними, особливо при великій кількості фільтрів та великому розмірі зображень.
- Схильність до перенавчання - якщо ви використовуєте невелику кількість даних або маєте складну модель, CNN може стати схильним до перенавчання.

1.4.4 Розпізнавання за допопомогою попередньо навчених моделей

Однією із перших спроб використання глибокого навчання для розпізнавання символів є використання архітектури LeNet-5. Це згорткова нейронна мережа яка була створена для розпізнавання рукописних цифр на зображеннях формату 32 х 32 пікселів. Вона має 7 шарів, та використовувалася кількома банками для розпізнавання рукописних цифр на чеках, в якості навчальної вибірки слугував набір MNIST. Ця архітектура показала найвищу точність класифікації рукописних чисел серед доступних тоді рішень у 1998 році. Це сприяло подальшому розвитку використання глибокого навчання у штучних нейронних мережах. Наразі ця архітектура вважається застарілою, і на заміну їй створено більш ефективні архітектури.

AlexNet, GoogleNet (Inception), VGG (Visual Geometry Group), та ResNet (Residual Network) - це чотири різні архітектури глибоких нейронних мереж, які використовуються для завдань обробки зображень, зокрема для класифікації зображень. Розглянемо їхні особливості:

AlexNet (створена українським розробником):

- Перша глибока нейронна мережа, що виграла конкурс з класифікації ImageNet.

- Складається з п'яти згорткових шарів та трьох повністю з'єднаних шарів.
- Використовує функцію активації ReLU та шари Local Response Normalization (LRN).
- Велика кількість параметрів.

GoogleNet (Inception):

- Використовує модуль Inception, який включає в себе паралельні згорткові шари різних розмірів та пулінг.
- Зменшує кількість параметрів, використовуючи Global Average Pooling.
- Застосовуєся концепція "блоків" для підвищення глибини мережі.

VGG:

- Мережа складається з великої кількості невеликих згорткових шарів (до 19 шарів).
- Всі згорткові шари мають невеликий розмір фільтрів (3x3) та крок 1.
- Використовує тільки згорткові та повністю з'єднані шари.

ResNet:

- Вперше використовує концепцію "залишкових" (residual) блоків, де вхідні дані додаються до виходу після згорткового шару.
- Може мати дуже глибоку структуру (понад 100 шарів) без проблем з зниканням або вибуханням градієнтів.
- Дозволяє тренування дуже глибоких мереж та приніс значні покращення у швидкості та точності.

Навчання цих моделей проводилося на наборі даних ImageNet - це один з найбільших та найважливіших наборів даних у галузі комп'ютерного зору та машинного навчання. Він включає в себе велику кількість зображень, які призначені для тренування та валідації моделей нейронних мереж для завдань класифікації об'єктів. Основні характеристики ImageNet:

- Кількість зображень: 14,197,122 зображень.В завданнях ImageNet Large Scale Visual Recognition Challenge (ILSVRC) містить більше 1 мільйона зображень для навчання та валідації.
- Кількість класів: основний набір даних має близько 1 000 класів об'єктів, наприклад, коти, собаки, автомобілі, люди та інші об'єкти.
- Розміри зображень: зазвичай зображення в наборі мають різні розміри, але зазвичай вони розміром більше 200x200 пікселів.
- Розподіл об'єктів: ImageNet містить різноманітні об'єкти та сцени, що робить його репрезентативним для різноманітних завдань в галузі комп'ютерного зору.

В таблиці наведено порівняльну характеристику даних архітектур, які навчалися на даному наборі.

Таблиця 1.1

Порівняння попередньо навчених алгоритмів

	Кількість нейронів	Кількість шарів	Точність
AlexNet	60 млн.	8	близько 58%
GoogleNet	Декілька млн.	22	Більше 70-75%
VGG	до 138 млн. для VGG-19)	Від 16 до 19	близько 70-75%
ResNet	152 млн.	Більше 100	Може перевищувати 95%

Різні дослідження, показують що на точність розпізнавання зображень досить сильно впливає складність архітектури моделі, що можна побачити в таблиці 1. Але використання цих архітектур вимагає досить багато обчислювальних ресурсів та часу для їх навчання, тому для їх застосування

використовуються попередньо навчені моделі. Що стосується складності використовуваних архітектур CNN, на думку деяких дослідників менш складні моделі, менш точні, але мають вищу точність класифікації, та навчання, а моделі навчені з нуля, як правило показують кращі результати розпізнавання зображень.

2 ДОСЛІДЖЕННЯ ПРОБЛЕМИ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

Однією з основних проблем при навчанні моделей для розпізнавання зображень, є залежність від якості навчальної вибірки. Тобто, точність методу розпізнавання зображень залежить від того, на якому наборі даних він був навчений. Якщо навчальна вибірка не є репрезентативною для набору даних, на якому буде здійснюватися розпізнавання, то точність методу може істотно знизитися. Сюди ж відноситься і розмір навчальної вибіки, як показує практика кількість навчальних зразків сильно впливає на якість навчання моделі.

Ці проблеми можуть призвести до таких наслідків:

- Перенавчання - нейромережа може навчитися відтворювати випадкові особливості в навчальних даних, а не загальні закономірності. Це може призвести до зниження точності нейромережі на тестових даних.
- Нестача загальності - нейромережа може навчитися розпізнавати лише певні види зображень, а не всі можливі види. Це може призвести до зниження точності нейромережі на зображеннях, які відрізняються від навчальних даних.

2.1 Аналіз архітектури моделей для розпізнавання зображень та визначення їх критичних точок

Кожна архітектура має свої особливості, та обмеження. Для більш якісного представлення проблеми, було здійснено аналіз методів розпізнавання зображень, та проведено дослідження залежності результатів розпізнавання зображень від об’єму навчальної вибірки.

2.1.1 Метод SVM

Метод опорних векторів (Support Vector Machine або SVM) — це алгоритм машинного навчання для задач класифікації та регресії. Його можна використовувати і для розпізнавання зображень в завданнях бінарної чи багатокласової класифікації. Основна ідея полягає в пошуку оптимального гіперплощинного розділення між класами в просторі ознак.

Архітектура SVM:

Вибір ядра (Kernel) - SVM використовує ядра для відображення ознак у вищорозмірний простір, де дані можуть бути лінійно розділені. Популярні ядра включають лінійне, поліноміальне та гаусівське (RBF) ядра.

Оптимізаційна задача - задача SVM полягає в знаходженні оптимальної гіперплощини, яка максимізує відстань між класами (маржа) і мінімізує помилки класифікації. Це формалізується як оптимізаційна задача (2.1).

$$min_{w,b} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} max\,(0,1 - y_i(w \cdot x_i + b)) \quad (2.1)$$

де w — ваговий вектор;

b — зсув (bias);

C — параметр регуляризації;

x_i— вектор ознак і-го зразка;

y_i — мітка класу для і-го зразка.

Опорні вектори - під час оптимізації обчислюються опорні вектори — зразки даних, які мають найбільший вплив на визначення гіперплощини.

Алгоритм SVM:

1. Навчання:
   - Подаємо навчальні дані на вхід.

- Вибираємо ядро та параметри ядра.
- Розв'язуємо оптимізаційну задачу для знаходження параметрів w та b.

2. Прогнозування:

- Підставляємо нові дані у визначену гіперплощину.
- Визначаємо клас на основі знака виразу $w \cdot x + b$.

Для представлення результатів класифікації за допомогою моделі SVM було використано метод головних компонент (PCA, Principal Component Analysis). PCA це статистичний метод зменшення розмірності, який використовується для виявлення основних шаблонів в наборі даних. Головна ідея PCA полягає в перетворенні високорозмірних даних у набагато менший обсяг, зберігаючи при цьому якнайбільше можливої інформації.

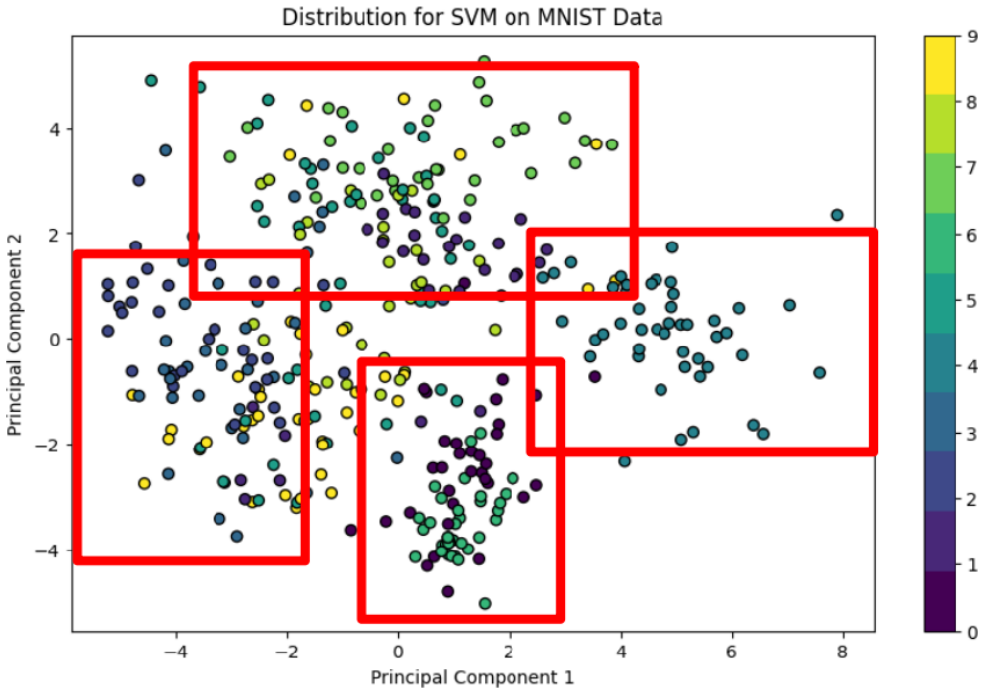

Рис.2.1 Представлення класифікації SVM на основі PCA

Графік розподілу для SVM на основі методу головних компонент (PCA) відображає прогнозовані класи для тестового набору даних MNIST в двовимірному просторі головних компонент. Кожен кольоровий пункт на графіку представляє один екземпляр тестового набору даних, при цьому кольори відповідають різним класам цифр (від 0 до 9). Точність розпізнавання на тестовому наборі даних склала 97.50%

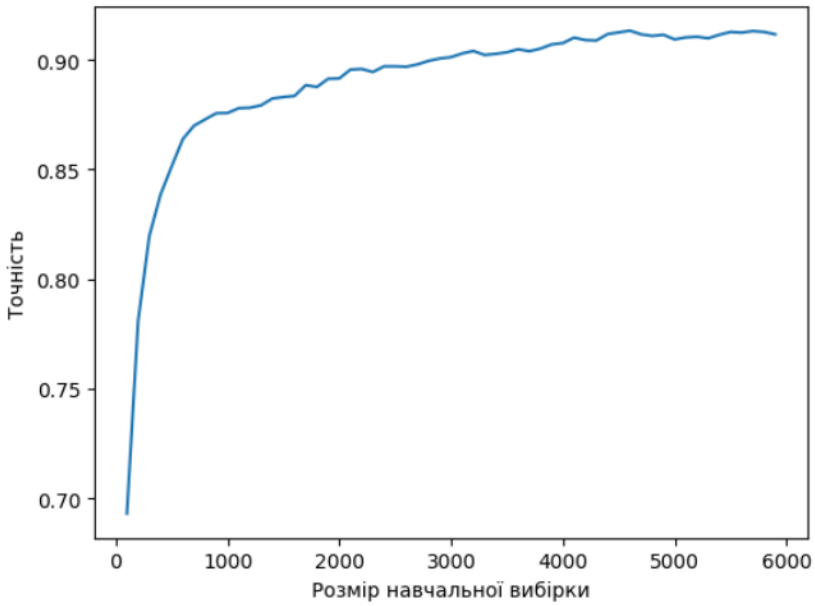

Рис. 2.2 Графік залежності методу SVM від навчальної вибірки

Як видно з графіка, точність розпізнавання зростає зі збільшенням розміру навчальної вибірки. Це пояснюється тим, що з більшою навчальною вибіркою модель може краще навчитися розділяти дані різних класів. Однак, точність розпізнавання не може зростати нескінченно. Це пов'язано з тим, що завжди буде існувати деякий шум у даних, який неможливо повністю усунути.

У цьому випадку точність розпізнавання досягає свого максимуму при розмірі навчальної вибірки близько 4000. При подальшому збільшенні розміру навчальної вибірки точність залишається приблизно на одному рівні.

2.1.2 Метод CNN

Згорткові нейронні мережі (CNN) — це тип нейронних мереж, які спеціалізуються на обробці зображень і використовують згорткові та пулінгові

шари для ефективного виявлення та ієрархічного вивчення ознак. Вони широко використовуються в області комп'ютерного зору, розпізнавання образів та інших схожих задачах.

Архітектура CNN

CNN складається з наступних шарів:

- Шар вхідного зображення: отримує на вхід зображення.
- Згорткові шари: застосовують фільтри до зображення, щоб витягти з нього особливості.
- Шари пулінгу: зменшують розмір зображення, зберігаючи при цьому основні особливості.
- Шари повнозв'язних нейронів: перетворюють вихід згорткових шарів і пулінгу в вектор, який потім класифікується за допомогою логістичної регресії.

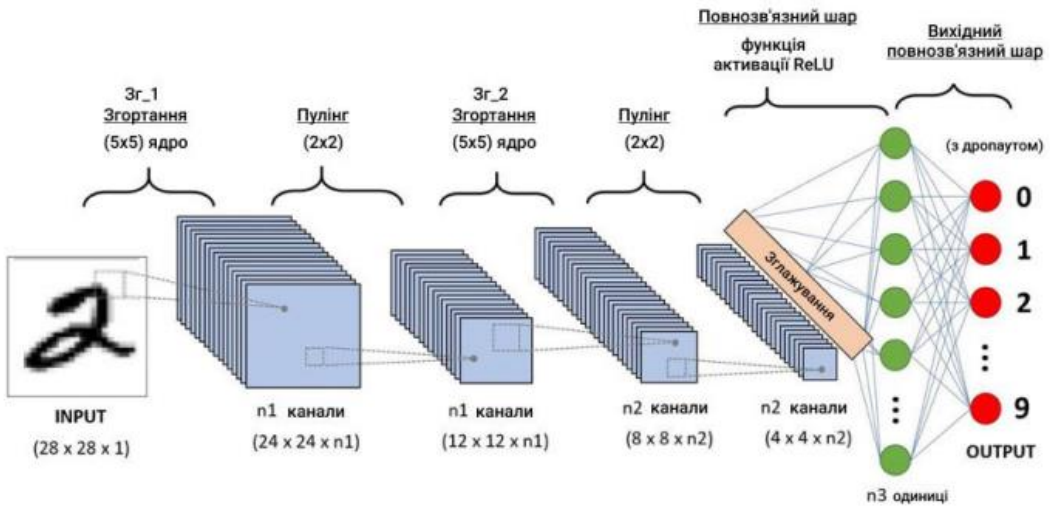

Рис. 2.3 Архітектура CNN

Алгоритм CNN

Основні етапи алгоритму CNN такі:

Введення (Input Layer):

- Зображення або вхідні дані представляються у вигляді матриці пікселів.
- Кожен піксель може бути розглянути як вхідний нейрон.

Згортка (Convolutional Layer):

- Використовуються фільтри (ядра), які рухаються по вхідному зображенню.
- Кожен фільтр виділяє певні особливості або шари вхідних даних, формуючи карту ознак.

Функція активації (Activation Function):

До виходу з фільтра застосовується нелінійна функція активації, така як ReLU (Rectified Linear Unit), для нелінійності та видалення негативних значень.

Згорткові шари і пулінг (Convolutional and Pooling Layers):

Декілька згорткових і пулінгових шарів може бути використано для виділення все більше абстрактних функцій та зменшення просторових розмірів.

Пулінг (Pooling Layer):

Зменшення розмірів карти ознак, використовуючи операції пулінгу, такі як MaxPooling або AveragePooling. Це допомагає зменшити кількість параметрів та обчислювальні витрати.

Повністю з'єднані шари (Fully Connected Layers):

Після відносно невеликої кількості згорткових та пулінгових шарів використовуються повністю з'єднані шари для класифікації або регресії. Повністю з'єднані шари обробляють абстрактні функції, отримані на попередніх етапах, для визначення вихідного результату.

Функція втрат (Loss Function):

Функція втрат L визначає різницю між прогнозованими значеннями y_{pred} та фактичними значеннями y_{true} і використовується для оцінки того, наскільки точно прогнози моделі відповідають даним (2.2):

$$L(y_{pred}, y_{true}) \tag{2.2}$$

Мета полягає в тому, щоб ця втрата була мінімальною. Залежно від типу завдання (класифікація, регресія і т. д.), використовуються різні функції втрат. Для багатокласової класифікації часто використовується крос-ентропійна функція втрати (2.3). Наприклад, для класифікації з трьома класами і вхідними векторами ймовірностей класів p та бінарних міток y:

$$CrossEntropyLoss = -\frac{1}{N}\sum_{i=N}^{N}\sum_{j=N}^{C} y_{ij}\log(p_{ij}) \tag{2.3}$$

де N - кількість прикладів у навчальному наборі;

C - кількість класів;

y – бінарні мітки;

p – ймовірності класів.

Оптимізація (Optimization):

Оптимізація в контексті нейронних мереж включає в себе процес адаптації ваг (параметрів) моделі так, щоб мінімізувати функцію втрат. Одним з найпоширеніших методів оптимізації є градієнтний спуск. Основні етапи цього процесу виглядають наступним чином:

Обчислення градієнтів:

- Визначення похідних функції втрати відносно ваг.
- Градієнти вказують напрямок та розмір змін для кожного параметра.

Оновлення ваг:

- Зменшення ваг в напрямку, протилежному градієнту, з метою мінімізації функції втрат.
- Крок навчання (learning rate) регулює розмір кроку при оновленні ваг.

Ітерації:

- Процес обчислення градієнтів та оновлення ваг повторюється чергову кількість разів (епох).
- Градієнтний спуск може мати різні модифікації, такі як стохастичний градієнтний спуск (SGD), метод моменту та інші, які покращують швидкість та ефективність оптимізації.

Процес градієнтного спуску включає обчислення градієнта функції втрат L відносно параметрів моделі $\theta(2.4)$:

$$\nabla_\theta L(\theta) \tag{2.4}$$

де $\nabla\theta$ - оператор градієнта.

Після обчислення градієнта, ваги оновлюються відповідно до наступної формули(2.5):

$$\theta_{\text{нові}} = \theta_{\text{старі}} - \eta\nabla_\theta L(\theta_{\text{старі}}) \tag{2.5}$$

де η - крок навчання, який визначає розмір кроку алгоритму оптимізації;

θ – параметри моделі.

Цей процес повторюється протягом кількох ітерацій (епох) для того, щоб мінімізувати функцію втрат і навчити параметри моделі. У практиці часто використовуються більш складні методи оптимізації, такі як Adam, RMSprop, але

основна ідея залишається схожою - зменшення функції втрат за допомогою адаптивного оновлення ваг.

Загальна ідея полягає в тому, щоб ваги моделі навчалися таким чином, щоб при прогнозуванні вони мінімізували помилку моделі на тренувальних даних і, в той же час, залишалися загальні для нових, раніше не бачених даних (уникнення перенавчання).

Навчання (Training):

Проведено експерементальне розпізнавання зображень за допомогою моделі CNN, для демонстрації залежності точності розпізнавання від розміру навчальної вибірки. Модель тренується на тренувальних даних, а потім перевіряється на валідаційних та тестових даних для оцінки її ефективності.

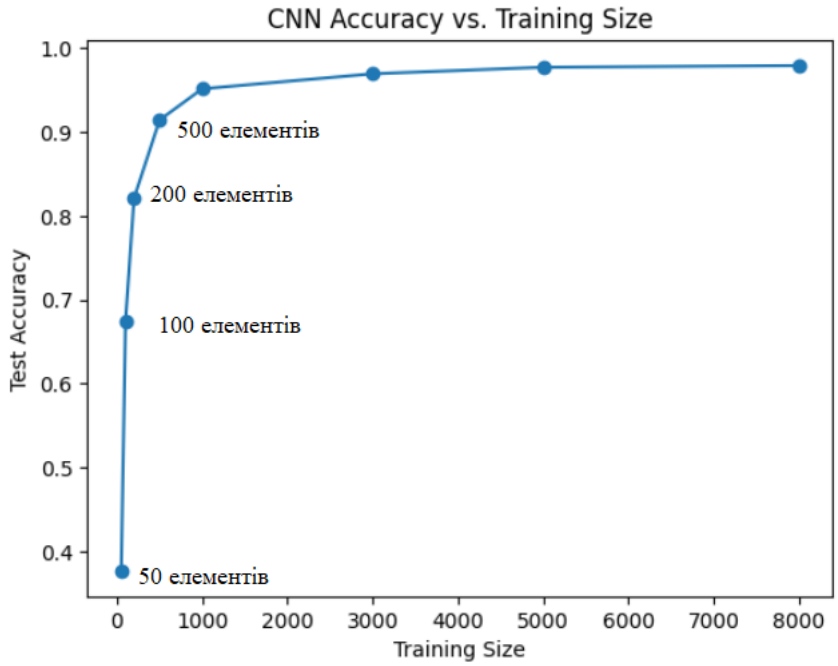

Рис. 2.4 Графік залежності точності CNN від розміру навчальної вибірки

За результатами навчання модель здатна розпізнавати зображення з тестового набору з точністю ±98%. На графіку продемонстровано залежність точності розпізнавання від розміру навчальної вибірки. В результаті можна зробити

висновок що медель найактивніше навчається до відмітки 1000 елементів, і здатна показати більше 95%.

2.1.3 Метод k-найближчих сусідів

Метод k-найближчих сусідів (k-NN) є одним з найпростіших інстанційних методів машинного навчання та використовується як для задач класифікації, так і для регресії. Основна ідея полягає в тому, що об'єкти з однаковими або схожими характеристиками зазвичай розташовані близько один від одного у просторі ознак.

Основні кроки алгоритму k-NN включають наступне:

Вибір значень k:

Визначаємо кількість сусідів (k), яку хочемо використовувати для прийняття рішення. Це число повинно бути непарним, щоб уникнути випадків "голосування" з однаковою кількістю голосів.

Визначення метрики відстані:

Позначимо набір даних як (2.6):

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \quad \text{.....(2.6)}$$

де x_i - вхідні ознаки об'єкта;

y_i - відповідна мітка (клас для класифікації або значення для регресії).

Нехай x - новий тестовий об'єкт, для якого ми хочемо зробити прогноз.Вибираємо метрику відстані $d(x_i, x)$, яка вимірює відстань між об'єктами x_i та x. Приклади метрик: евклідова відстань, Манхеттенська відстань, косинусна відстань тощо.

Обчислення відстані:

Обчислюємо відстані (2.7) між тестовим об'єктом x і всіма об'єктами у тренувальному наборі:

$$d_i = d(x_i, x) \text{ для } i = 1,2,\dots,N. \quad (2.7)$$

де $d -$ відстань.

Сортування за відстанями:

Сортуємо індекси об'єктів у тренувальному наборі за зростанням відстаней (2.8):

$$i_1, i_2, \dots, i_N, \text{ де } d_{i1} \leq d_{i2} \leq \dots \leq d_{iN} \quad \dots\dots\dots\dots\dots\dots (2.8)$$

Вибір k найближчих сусідів:

Вибираємо перші k об'єктів з відсортованого списку, отримуючи набір (2.9).

$$N_k = \{i_1, i_2, \dots, i_k\} \quad (2.9)$$

де N_k – список об’єктів.

Голосування або середнє значення:

У випадку класифікації обчислюємо голоси за кожен клас серед об'єктів N_k, і вибираємо клас, який отримав найбільше голосів.

Прийняття рішення:

Вивід прогнозу або класифікації на основі голосування або середнього значення.

Оцінка результату:

Оцінка якості моделі за допомогою метрик, таких як точність (у випадку класифікації) або середньоквадратична помилка (у випадку регресії).

Алгоритм k-NN простий та легко зрозумілий, але він може бути обчислювально витратним для великих наборів даних. Крім того, важливо правильно вибрати значення k та підібрати підходящу метрику відстані в залежності від характеру даних.

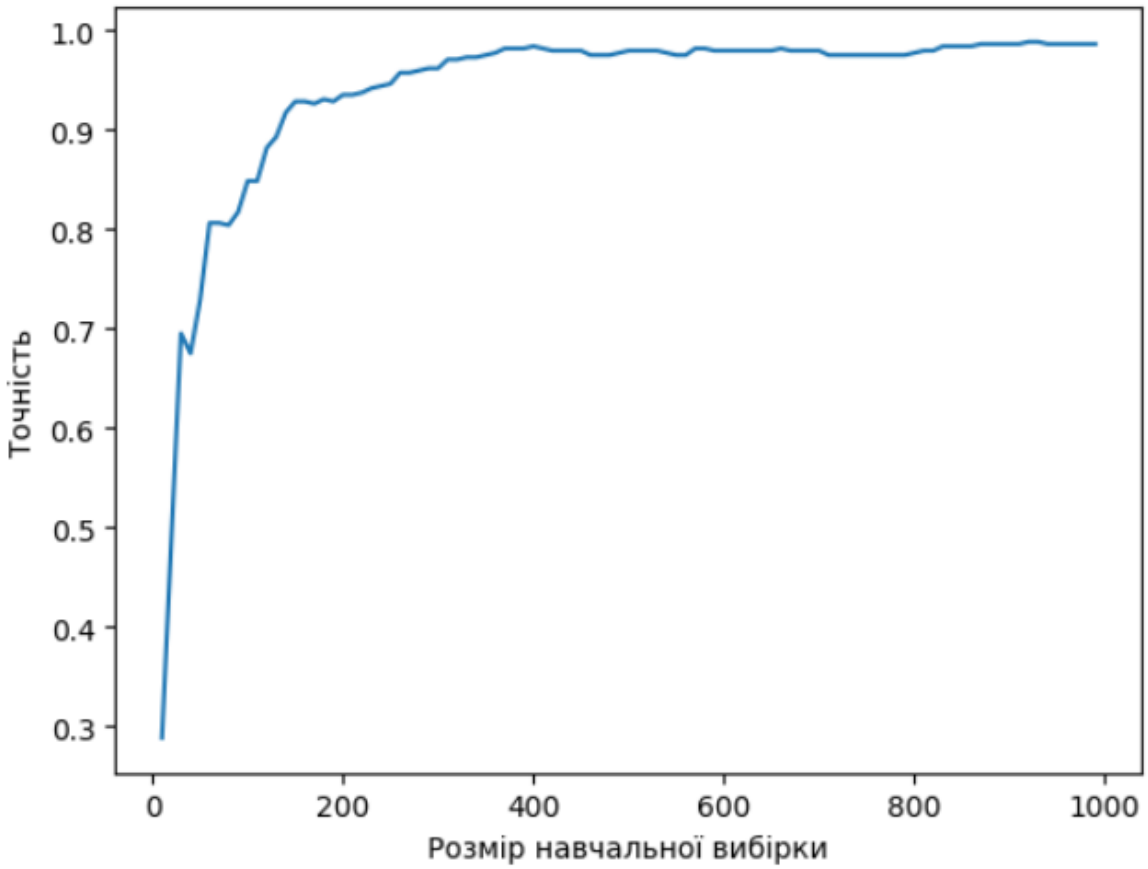

Рис. 2.5 Графік залежності точності розпізнавання K-NN
від розміру навчальної вибірки

Як видно з результатів, точність класифікатора K-NN зростає зі збільшенням розміру навчальної вибірки. Це пов'язано з тим, що при збільшенні розміру навчальної вибірки класифікатор має більше даних для навчання і може краще

ідентифікувати закономірності в наборі даних. У цьому випадку максимальна точність класифікатора становить близько 95%. Цей результат є досить високим і може бути використаний для практичних цілей.

2.2 Методи генерації зображень

Застосування кожного з вище наведених методів передбачає навчання моделі на навчальному наборі даних, та перевірки ефективності на валідаційному та тестовому наборі даних. Для завдання розпізнавання зображень навчальний набір грає важливу роль, адже ці дані прямо впливають на точність розпізнавання тестових зразків. При недостатній кількості зразків в навчальному наборі стикаємося з проблемами перенавчання чи недонавчання моделі, що впливає на її ефективність.

2.2.1 Бібліотека Image Data Generator

Для вирішення цієї проблеми, можна провести аугментацію навчальної вибірки за допомогою Image Data Generator з пакету tensorflow. При використанні даного методу, до зображень застосовується випадкове обертання, випадкове масштабування та витягування/стиск уздовж випадкової осі. Кількість згенерованих зображень може сягати в межах від 2 до 32 на кожен зразок.

Переваги Image Data Generator:

- Збільшує кількість даних для навчання. Image Data Generator може генерувати нові зображення з існуючих даних. Це може бути корисно, якщо ви маєте невеликий набір даних для навчання.
- Збільшує різноманітність даних. Image Data Generator може генерувати зображення з різними змінами, такими як поворот, масштабування та зміна

освітлення. Це може допомогти моделі навчитися краще розпізнавати об'єкти в різних умовах.

- Зменшує перенавчання. Image Data Generator може генерувати зображення з шумом або іншими артефактами. Це може допомогти моделі стати менш сприйнятливою до шуму в реальних даних.

Недоліки Image Data Generator:

- Може бути повільним. Генерація нових зображень може бути трудомістким процесом.
- Може бути складним у налаштуванні. Image Data Generator може вимагати деяких налаштувань, щоб генерувати зображення, які є корисними для навчання.
- Може створювати нереалістичні зображення. Image Data Generator може генерувати зображення, які не є реалістичними. Це може призвести до того, що модель буде погано працювати з реальними даними.

В цілому, Image Data Generator є непоганим інструментом, який може бути використаний для поліпшення ефективності моделей розпізнавання зображень. Але основна проблема в тому, що він не може створити унікальних зразків, такі як рукописні символи. Для цього завдання необхідно підбирати інший підхід.

Існують різні методи для генерації зображень, і вони можуть використовувати різні підходи та алгоритми. Згідно з таксономією OpenAI існують три популярні типи генеративних мереж: VAE, GAN і авторегресивні мережі.

2.2.2 Генеративно-суперечлива мережа (GAN)

GAN, або генеративно-суперечлива мережа (Generative Adversarial Network), є типом штучних нейронних мереж, який введений Іаном Гудфеллоу та його колегами в 2014 році.

Генеративно-суперечлива мережа (GAN) складається з двох головних компонентів: генератора та дискримінатора. Позначимо випадковий вектор, який є вхідним шумом для генератора, як z. Зображення, яке генерується генератором, будемо позначати як $G(z)$. Дискримінатор приймає на вхід зображення і видає ймовірність того, що воно є справжнім (наприклад, знаходиться у тренувальному наборі) проти ймовірності того, що воно є сгенерованим (від генератора). Позначимо ймовірність того, що дискримінатор визначить зображення як справжнє як $D(x)$, де x - справжнє зображення.

Математично, процес тренування GAN можна представити наступним чином:

- Генератор (G) намагається максимізувати ймовірність того, що дискримінатор помиляється (2.10). Тобто, генератор намагається максимізувати:

$$log(D(G(z))) \rightarrow max \tag{2.10}$$

де $G(z)$ - зображення, яке генерується генератором;

$D(x)$ - ймовірність того, що дискримінатор визначить зображення як справжнє.

- Дискримінатор намагається правильно відрізнити справжні зображення від сгенерованих(2.11). Тобто, дискримінатор намагається максимізувати

$$log(D(x)) + log(1 - D(G(z))) \rightarrow max \tag{2.11}$$

де $G(z)$ - зображення, яке генерується генератором;

$D(x)$ - ймовірність того, що дискримінатор визначить зображення як справжнє.

Отже, функція втрат GAN може бути записана як (2.12):

$$min_G\ max_D\ V(D,G) = E_{x\sim p_{data}(x)}[log(D(x))] + E_{z\sim p_z(z)}[log(1 - D(G(z)))] \tag{2.12}$$

де $p_{data}(x)$ - розподіл справжніх даних;

$p_z(z)$ - розподіл шуму.

Процес тренування включає в себе ітеративну гру між генератором і дискримінатором, де кожен намагається покращити свою власну функцію втрат, що призводить до збалансованого стану, де генеровані зображення стають все більш схожими на справжні.

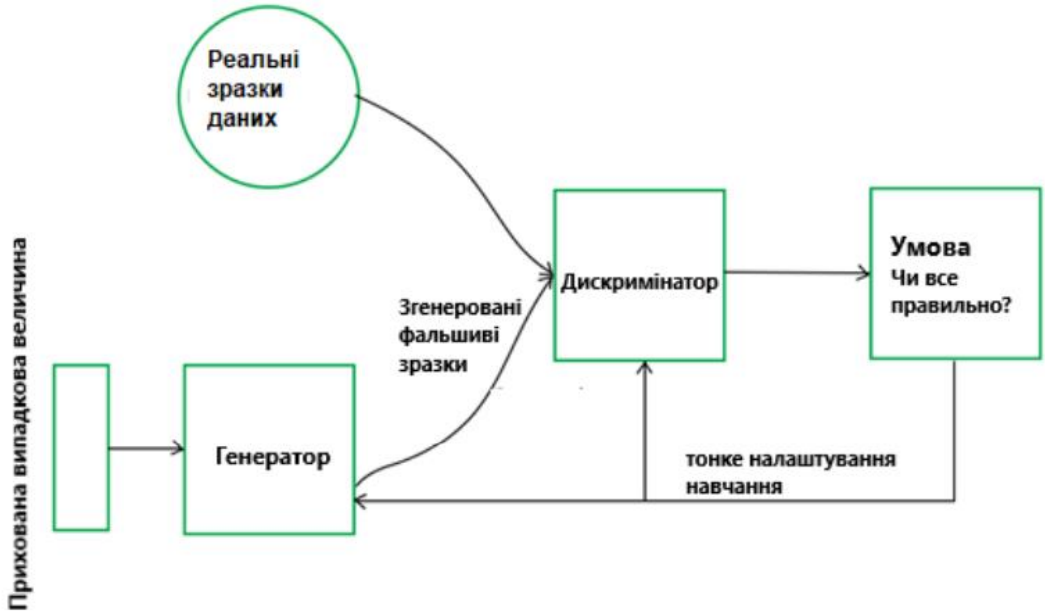

Рис. 2.6 Алгоритм роботи GAN

Переваги GAN:

- Генерація високоякісних зображень: GAN може створювати зображення високої якості, які часто непотрібно відрізняти від справжніх.

- Різноманітність генерації: GAN може генерувати різноманітні зображення, що робить його ефективним для творчих завдань, таких як генерація мистецтва або дизайну.
- Без навчання з учителем: GAN може працювати без навчання з учителем, тобто без використання явно позначених прикладів.

Недоліки GAN:

- Нестабільність навчання: Процес навчання GAN може бути нестабільним, і іноді він може зіштовхуватися з проблемою зникнення та вибуху градієнта.
- Можливість перенавчання: GAN може вивчити занадто конкретні особливості тренувальних даних і стати непридатним для генерації нових, реалістичних зображень.
- Час та обчислювальні витрати: Навчання GAN може вимагати значних обчислювальних ресурсів та часу.

Незважаючи на недоліки, GAN залишається потужним інструментом для завдань генерації зображень та використовується в різних областях, таких як комп'ютерне зорове сприйняття, медичне зображення, графічний дизайн і багато інших.

2.2.3 Варіаційний автокодер (VAE)

Варіаційний автоенкодер (VAE) — це генеративна модель, яка вивчає розподіл у латентному просторі даних. У VAE є дві основні частини: енкодер і декодер. Модель навчається апроксимувати розподіл вхідних даних у латентному просторі, і потім може генерувати нові зображення, випадковим чином вибираючи точки у цьому просторі. Він також може використовуватися для генерації нових варіацій інсайтів в існуючих даних.

Архітектура VAE

Кодер (Encoder): повністю з'єднана нейронна мережа, яка приймає вхідні дані і видає параметри гаусівського розподілу(2.13). Вхідне зображення x кодується у латентний простір за допомогою параметрів гаусівського розподілу

$$q_\phi\,(z \mid x) = N\left(z \,\middle|\, \mu_\phi\,(x), \sigma_\phi\,(x)\right) \tag{2.13}$$

де $\mu_\phi\,(x)$ це середнє відхилення;

$\sigma_\phi\,(x)$ - та стандартне відхилення, що залежать від вхідного зображення x. Цю мережу також іноді називають "енкодером".

Латентний простір:

Латентний простір є простором, у якому знаходяться латентні представлення даних. Це гаусівський розподіл, параметри якого визначаються кодером. Латентний вектор z зазвичай згенерований, використовуючи середнє значення та логарифм дисперсії, які отримані від кодера, а також випадковий шум.

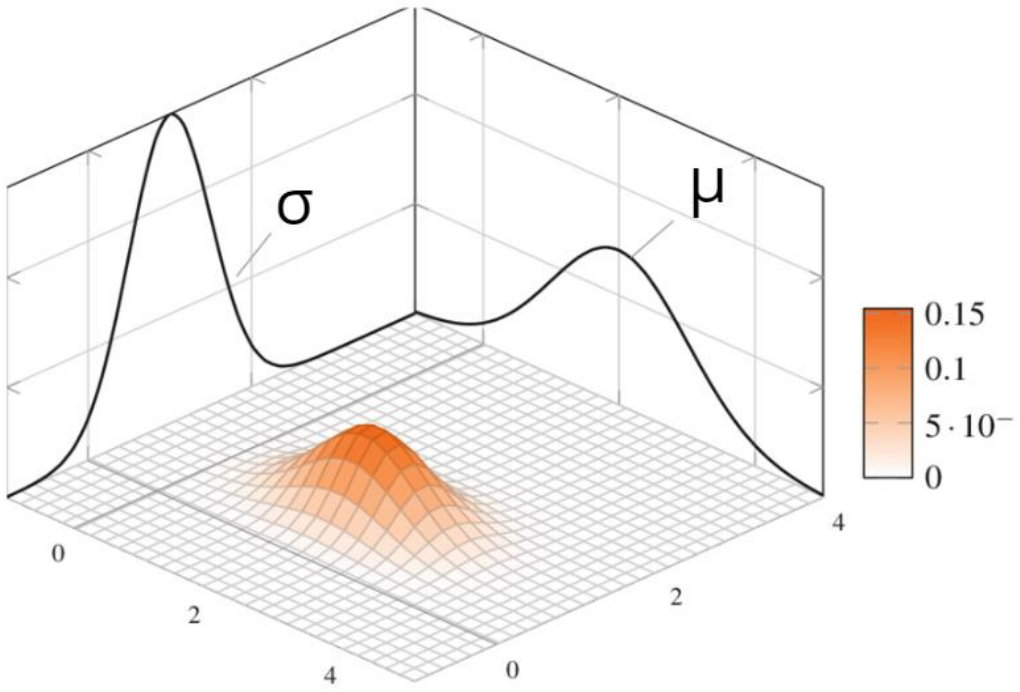

Рис. 2.7 Представлення латентного простору

Декодер (Decoder):

Декодер відповідає за відновлення вхідних даних з латентного простору(2.14). Він приймає згенеровані латентні представлення та намагається відтворити вхідні дані.

$$p\theta(x' \mid z) = N\big(x' \mid \mu_\theta(z), \sigma_\theta(z)\big) \tag{2.14}$$

де $\mu_\phi\ (x)$ це середнє відхилення;

$\sigma_\phi\ (x)$ - та стандартне відхилення, що залежать від залежать від латентного вектора z.

Декодер також є повністю з'єднаною нейронною мережею, що приймає на вхід вектор латентного простору і видає реконструйовані дані.

Функція втрат:

Для оцінки ефективності моделі використовується ELBO, що оцінює різницю між точнішим значенням логарифма ймовірності тестового зображення та нижнім обмеженням цієї ймовірності (2.15).

$$\log p_\theta\,(\mathrm{x}) \geq \mathrm{E}_{\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})}[\log p_\theta\,(\mathrm{x} \mid \mathrm{z})] - \mathrm{D}_{\mathrm{KL}}(\mathrm{q}_\phi(\mathrm{z} \mid \mathrm{x}) \,||\, \mathrm{p}(\mathrm{z})) \tag{2.15}$$

де D_{KL} - дивергенція Кульбака-Лейблера;

$\mathrm{p}(\mathrm{z})$ - апріорний розподіл латентних векторів.

Загальна функція втрат VAE об'єднує ELBO з додатковою регуляризацією, забезпечуючи близькість розподілу в латентному просторі до апріорного розподілу (2.16).

$$\mathcal{L}(\theta, \phi; \mathrm{x}) = -\mathrm{E}_{\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})}[\log p_\theta\,(\mathrm{x} \mid \mathrm{z})] + \mathrm{D}_{\mathrm{KL}}(\mathrm{q}_\phi(\mathrm{z} \mid \mathrm{x}) \,||\, \mathrm{p}(\mathrm{z})) \tag{2.16}$$

де $\mathcal{L}$ - це функція втрат VAE;

θ і ϕ - параметри моделі декодера і енкодера відповідно.

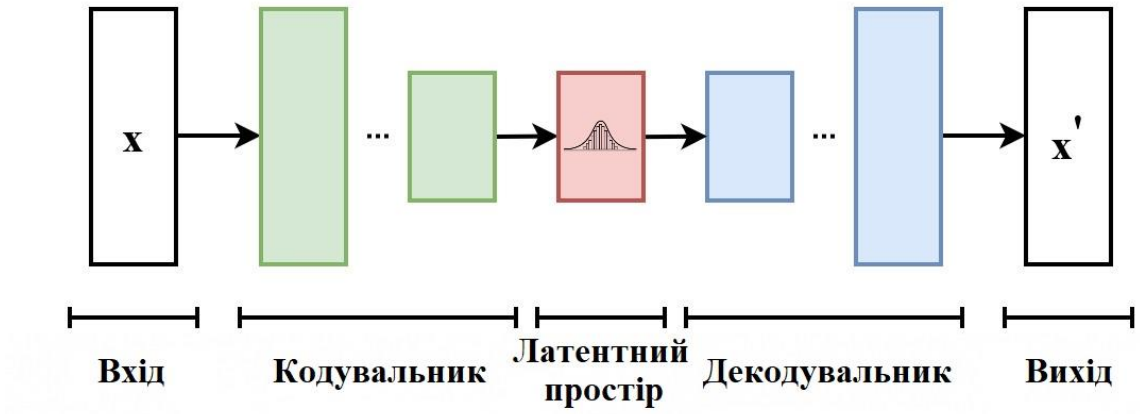

Рис. 2.8 Архітектура VAE

Переваги варіаційного автокодувача (VAE):

- Генерація нових даних – VAE дозволяє генерувати нові приклади даних, які подібні до тих, які були використані для навчання моделі. Це робить його корисним для завдань генерації контенту.
- Латентний простір – використання латентного простору дозволяє відділити основні характеристики даних, що може полегшити аналіз та розуміння внутрішніх представлень.
- Застосування до різних типів даних – VAE може бути використаний для роботи з різними типами даних, включаючи зображення, текст та числові дані.
- Гнучкість – можливість використовувати різні архітектури та варіанти модифікацій, дозволяє адаптувати VAE до конкретних завдань.

Недоліки варіаційного автокодувача (VAE):

- Чутливість до параметрів – VAE чутливий до налаштувань параметрів, таких як розмірність латентного простору, та може вимагати хорошої оптимізації для досягнення найкращих результатів.

- Реконструкція може бути неідеальною – на практиці реконструкція може бути не завжди ідеальною, особливо при великих розмірностях вхідних даних.
- Труднощі з оптимізацією – оптимізація великої кількості параметрів VAE може бути витратною з обчислювальної точки зору, і це може бути проблемою для великих об'ємів даних.
- Зміна форми розподілу – VAE передбачає гаусівський розподіл в латентному просторі, що може бути недостатнім для деяких складних структур даних.
- Дивергенція Кульбака-Лейблера – враховувати дивергенцію Кульбака-Лейблера в обчисленнях може бути обчислювально витратною задачею.

2.2.4 Авторегресивні мережі

Авторегресивні мережі **(**AR-мережі**)** для генерації зображень використовують концепцію авторегресії для послідовного синтезу пікселів або блоків пікселів у зображеннях. Основною ідеєю є те, що кожен піксель або група пікселів генерується на основі попередніх пікселів у зображенні AR-мережі для генерації зображень складаються з декількох шарів нейронів, які обробляють попередні пікселі, щоб зробити прогноз.

Авторегресивна модель порядку $AR(p)$ використовує p попередніх значень для прогнозування наступного значення (2.18). Математично це можна виразити як:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t \tag{2.18}$$

де X_t - це значення часового ряду у момент часу;

$\phi_1, \phi_2, \ldots, \phi_p$ - параметри моделі;

$X_{t-1}, X_{t-2}, \ldots, X_{t-p}$ - попередні значення;

ε_t - білий шум.

Ось як працює AR-мережа для генерації зображень:

- Набір даних зображень розбивається на ряд рядків пікселів.
- AR-мережа навчається на наборі даних рядків пікселів.
- AR-мережа використовується для прогнозування наступного пікселя в ряду пікселів.

Процес повторюється, поки не буде створено все зображення.

Переваги авторегресивних мереж (AR):

- Простота та інтерпретованість - AR-моделі мають просту математичну формулу та є легкими для інтерпретації, що робить їх привабливими для застосування в контексті, де важлива зрозумілість результатів.
- Ефективність на стаціонарних рядах - якщо часовий ряд є стаціонарним (не залежить від часу), AR-моделі можуть бути ефективними для прогнозування майбутніх значень.
- Гнучкість залежно від порядку (p) - збільшення порядку AR-моделі може дозволити більш точно враховувати складніші залежності в часовому ряді.

Недоліки авторегресивних мереж:

- Чутливість до стаціонарності - AR-моделі передбачають, що часовий ряд є стаціонарним, і можуть давати непралільні результати при роботі з нестаціонарними рядами.
- Неефективність на нестаціонарних рядах - для нестаціонарних часових рядів AR-моделі можуть бути менш ефективними, оскільки вони можуть не враховувати тренди та інші нелінійні зміни в часовому ряді.

- Залежність від правильного вибору порядку (p**)** - вибір оптимального порядку AR-моделі є нетривіальною задачею і може вимагати великої уваги до деталей та експериментів.
- Неспроможність робити прогнози для далекого майбутнього - AR-моделі можуть бути обмеженими у здатності робити точні прогнози для великого горизонту часу вперед, оскільки вони покладаються на попередні значення.

Авторегресивні мережі (Autoregressive Models) є класом моделей, які генерують дані шляхом прогнозування значень одного або декількох змінних на підставі попередніх значень. У контексті генерації зображень, авторегресивні моделі працюють, створюючи зображення піксель за пікселем чи блок за блоком. Тут є кілька прикладів авторегресивних мереж для генерації зображень:

- PixelRNN і PixelCNN — це моделі, які генерують зображення піксель за пікселем. PixelRNN використовує рекурентні нейронні мережі (RNN), а PixelCNN — згорткові нейронні мережі (CNN), для авторегресивного моделювання розподілу пікселів у зображенні. Вони дозволяють моделі враховувати контекст попередніх пікселів при генерації кожного нового пікселя.
- MADE(Masked Autoencoder for Distribution Estimation) — це авторегресивна модель, яка використовує ідею автоенкодера для навчання розподілу даних. Вона використовує маски, щоб забезпечити, що кожен елемент виходу залежить лише від попередніх елементів.
- NADE(Neural Autoregressive Distribution Estimator) — це авторегресивна модель, яка використовує нейронні мережі для моделювання умовних ймовірностей елементів вектора даних, виходячи з його попередніх елементів.
- WaveNet — призначений для генерації аудіоданих, також є авторегресивною мережею, що працює на основі умовних ймовірностей.

AR-мережі для генерації зображень працюють, використовуючи принципи машинного навчання, щоб навчитися на наборі даних зображень. Набір даних може бути будь-яким типом зображень, включаючи фотографії, малюнки та відео. Під час навчання AR-мережі намагаються знайти такі значення вагових коефіцієнтів, які дозволяють їй найкраще генерувати зображення, які схожі на зображення в наборі даних. AR-мережі для генерації зображень є потужним інструментом, який можна використовувати для створення нових і цікавих зображень. Однак вони мають деякі обмеження. Наприклад, AR-мережі можуть бути складними для навчання, і вони можуть бути чутливими до шуму в даних.

3 ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА МОДЕЛЮВАННЯ

3.1 Проектування автокодувальника для ефективної репрезентації рукописних зображень

Автокодувальники в самому простому варіанті - це нейронна мережа яка спочатку кодує вхідний сигнал в латентний простір, потім з цього стану знову розгортає та декодує дані в інший новий стан, формуючи вихідний сигнал. Розглянемо для прикладу найпростіший автокодувальник, з лінійною функцією активації

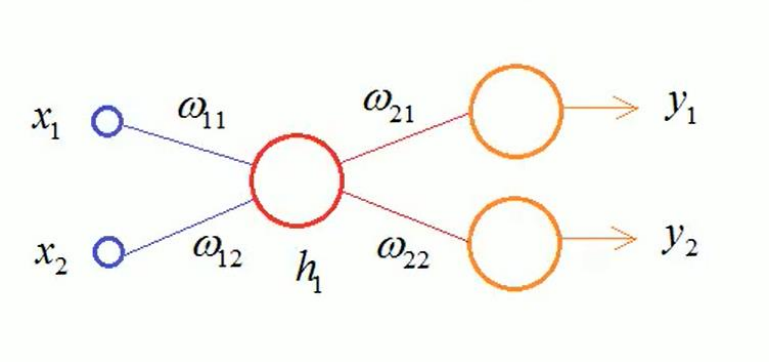

Рис. 3.1 Найпростіше представлення автокодувальника

В цій схемі кодер виконує дуже просту операцію, він формує число h_1 прихованого стану (3.1) на основі вектору x_1 x_2, використовуючи вагові коефіцієнти ω_{11} та ω_{12}.

$$h_1 = \omega_{11}x_1 + \omega_{12}x_2 \tag{3.1}$$

де $h -$ прихований стан;

ω – вагові коефіцієнти;

$x -$ вхідний сигнал.

В свою чергу декодер розгортає значення h_1 знову в двохвимірний вектор $y_1\, y_2$. (3.2)

$$\begin{cases} y_1 = h_1\, \omega_{21} \\ y_1 = h_1\, \omega_{21} \end{cases} \tag{3.2}$$

де $h -$ прихований стан;

ω – вагові коефіцієнти;

$y_1\, y_2$ $-$ вихідний сигнал.

Для представлення зображень цифр, з бази даних MNIST реалізуємо простий автокодувальник, у вигляді наступної повнозв'язної мережі. На вхід мережі ми будемо завантажувати рукописні цифри, а на виході будемо намагатися отримати точно такі ж рукописні цифри. Таким чином ми будемо кодувати вхідний сигнал в прихований стан, а потім декодувати отримуючи зображення.

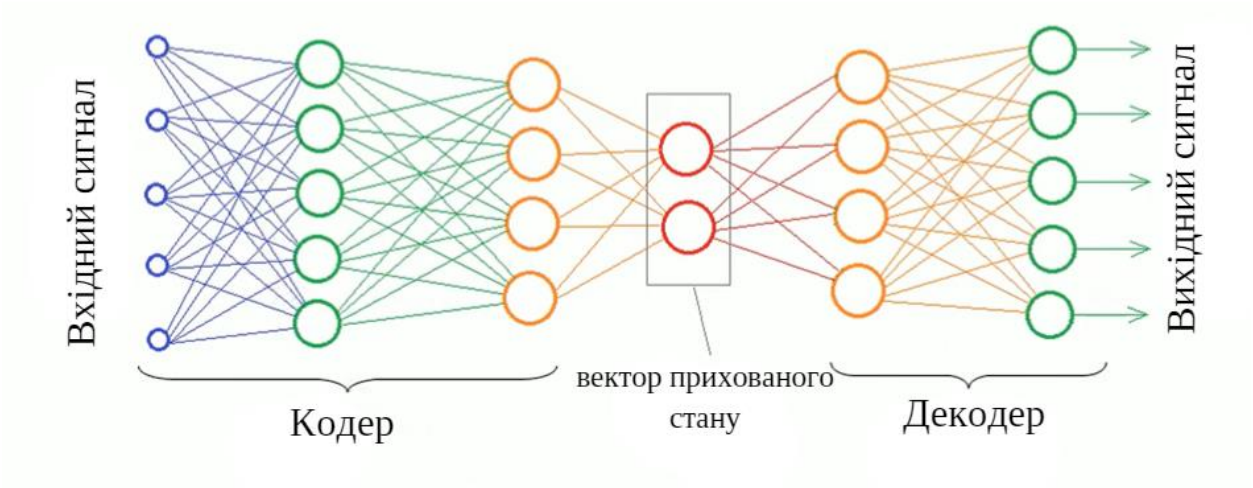

Рис. 3.2 Модель автокодувальника

Будь яке зображення розміром 28x28 пікселів, можна представити як точку 786 мірному просторі, більшість точок в цьому просторі будуть відповідати шумовим, незрозумілим зображення, і тільки невелика частина буде відповідати

цифрам. Кодер в процесі навчання намагається визначити область цифр в цьому багатомірному просторі. В результаті навчання на виході отримаємо множину точок двухмірних векторів які лежать в латентному просторі для тестового набору даних.

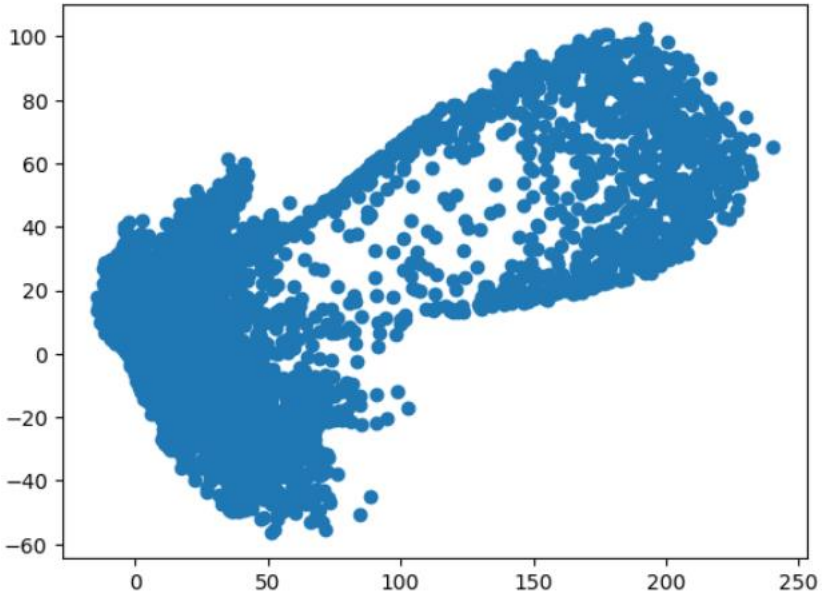

Рис. 3.3 Представлення латентного простору

Якщо ми будемо обирати точки в межах сформованої області, то можемо отримати якісь зрозумілі зображення. Для прикладу візьмемо вектор 50, -40 та відправимо його на декодер, на виході отримали цифру 7.

```
1  img = decoder.predict(np.expand_dims([50, -40], axis=0))
2  plt.imshow(img.squeeze(), cmap='gray')
```

```
1/1 [==============================] - 0s 19ms/step
<matplotlib.image.AxesImage at 0x7c7cd89d0d60>
```

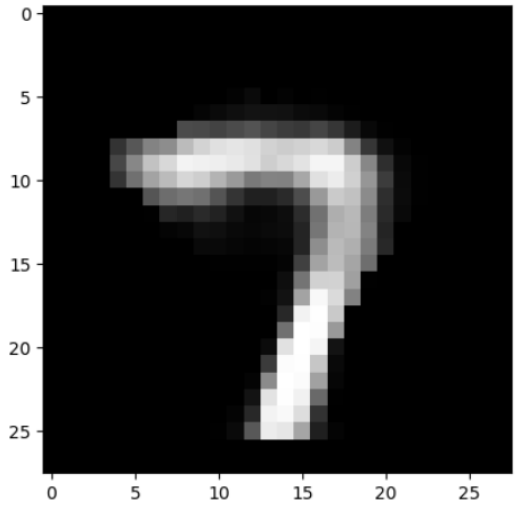

Рис. 3.4 Семплінг вектору в межах сформованої області

Експерементально на вхід декодера передамо вектор що виходить за межі сформованої області, для прикладу 250, 250 та отримуємо незрозуміле зображення, яке не належить до жодної з цифр.

```
1  img = decoder.predict(np.expand_dims([250, 250], axis=0))
2  plt.imshow(img.squeeze(), cmap='gray')
```

```
1/1 [==============================] - 0s 28ms/step
<matplotlib.image.AxesImage at 0x7c7cd884f130>
```

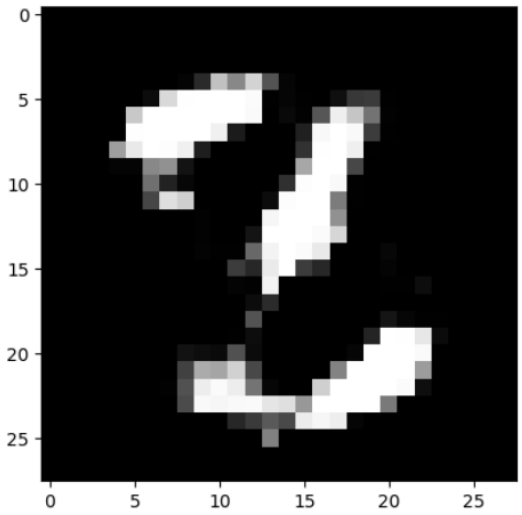

Рис. 3.5 Семплінг вектору за межами сформованої області

Таким чином виникає питання, які точки з латентного простору обирати, щоб отримати зображення саме цифр? Для вирішення цієї проблеми простір прихованих векторів повинен бути компактним, та представлений єдиною цілою областю.

3.1.1 Варіаційний автокодувальник

Для вирішення цієї задачі необхідно використовувати варіаційний автокодувальник. Варіаційний автокодувальник намагається сформувати область точок прихованого стану, згідно заданим законам розподілу. Часто обирається номальний розподіл (Гаусівський розподіл), так як він найбільш простий з обчислювальної точки зору, має просту форму, та повністю описується двома параметрами: середнє значення(або математичне очікування) та дисперсія.

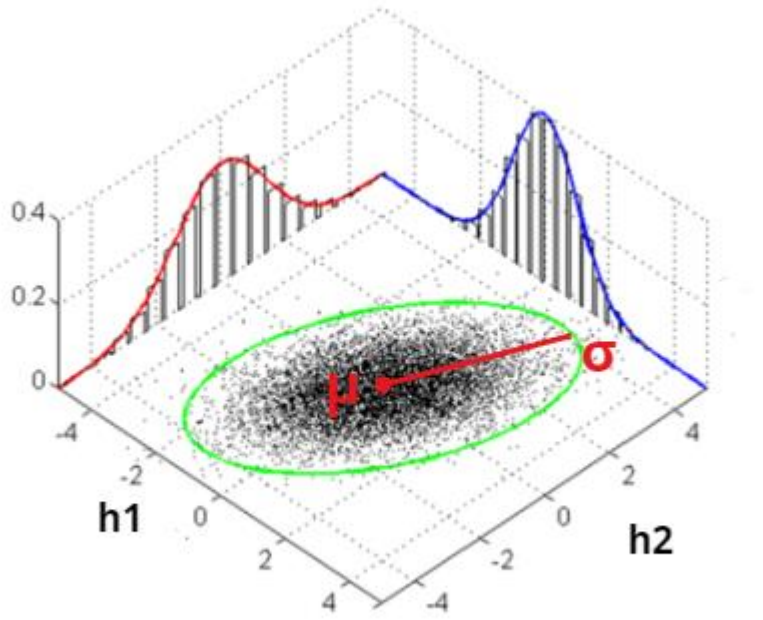

Рис. 3.6 Нормальний розподіл

Завдяки тому що варіаційний автокодувальник намагається згенерувати таку компактну область, можна мати впевненість що будь яка точка взята з розподілу, на виході декодеру буде видавати зрозумілі зображення. Тобто тут з'являється розуміння, як з латентного простору обирати точки, для генерації нових зображень. В цьому і є особливість варіаційних автокодувальників, та головна відмінність від звичайного автокодувальника. Отже ми маємо матиматичне очікування, дисперсію, сформований розподіл точок та бажаний розподіл (3.3).

$$h = (h_1, h_2)^T$$

$$\omega_G(h_1, h_2, \dots),$$

$$\omega_N(h_1, h_2, \dots) \tag{3.3}$$

де $\omega_G -$ сформований розподіл;

$\omega_N -$ номальний розподіл;

$h -$ прихований стан.

В процесі навчання мережі необхідно не тільки точно відтворити вхідний сигнал, але й розподілити точки латентного простіру якнайближче до нормального розподілу.

Для визначення міри розхоження між цими розподілами, необхідно визначити критерій якості, для цього існує такий критерій як дивергенція Кульбака-Лейблера (3.4). Для даного випадку відстань Кульбака-Лейблера визначається як:

$$D_{KL} = \frac{1}{2}(tr(\sum_G) + \mu_G^T \mu_G - k - \log(det\sum_G) \tag{3.4}$$

де $\mu_G -$ математичне очікування;

$tr(\sum_G) -$ сума значень головної діагоналі матриці;

$det\sum_G -$ детермінант матриці.

Матриця $\sum_G$ в свою чергу, це коваріаційна матриця вектору випадкових величин сформованого розподілу (3.5).

$$\sum_G = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ . & . & . & . \\ 0 & 0 & \dots & \sigma_k^2 \end{pmatrix} \tag{3.5}$$

де $\sigma_k^2 -$ дисперсія розподілу.

Матриця $\sum_N$, це коваріаційна матриця нормального розподілу, з нульовим математичним очікуванням, та одиничною дисперсією (3.6).

$$\sum_N = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ . & . & . & . \\ 0 & 0 & \dots & 1 \end{pmatrix} \tag{3.6}$$

На рисунку (3.7) зображено модель варіаційного автокодувальника, де відображено алгоритм побудови нормального розподілу латентного простору.

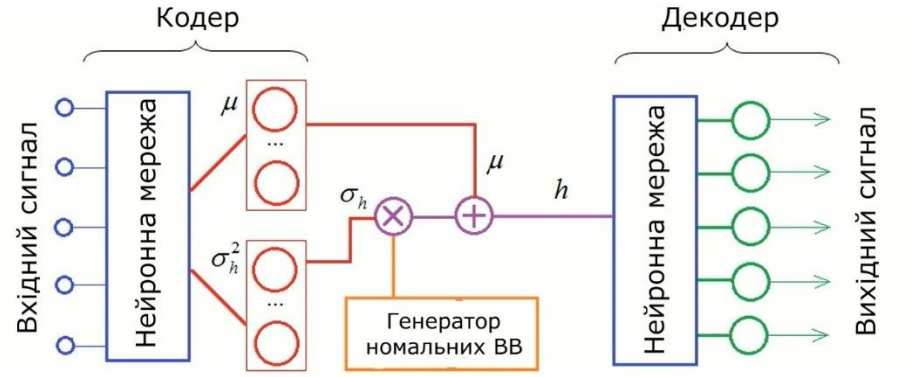

Рис. 3.7 Архітектура варіаційного автокодувальника

На виході кодера буде 2 вектора, $\mu -$ математичне очікування k нейронів, та $\sigma_h -$ k нейронів для дисперсії. На основі цих векторів буде генеруватися випадкова величина h(3.7):

$$h = \sigma_h \times N + \mu \tag{3.7}$$

де $\sigma_h -$ дисперсія,

$\mu -$ математичне очікування,

N – генератор номальних випадкових величин.

Таким чином, в процесі навчання нейрони кодера будуть формувати вектори математичного очікування та дисперсії, для кожного вхідного елементу x. Також в процесі навчання мінімізуються одразу 2 критерія. Перший критерій (3.8) необхідний для правильного відтворення сигналу, тобто так щоб вихідний сигнал намагався відтворити вхідний сигнал:

$$loss = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{3.8}$$

Де $x_i -$ вхідний сигнал,

$y_i -$ вихідний сигнал.

Другий критерій це якраз критерій Кульбака-Лейблера. Отже загальний критерій це сума цих двох критеріїв (3.9).

$$L = loss + D_{KL} \tag{3.9}$$

Отже після побудови моделі варіаційного автокодувальника, отримуємо правильний розподіл векторів в латентному просторі, тобто всі точки сформувалися з дисперсією наближеною ближче до нуля осі координат.

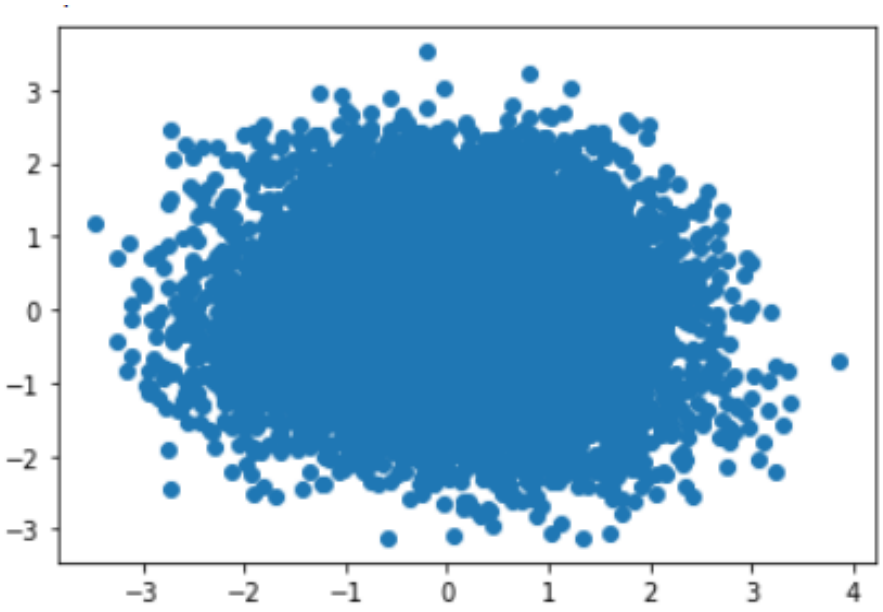

Рис. 3.8 – Нормальний розподіл латентного простору

Тепер ми можемо обирати будь які точки з даного розподілу, та отримувати зображеня цифр. Для демонстрації проведемо семплінг точок в квадраті від -3 до 3.

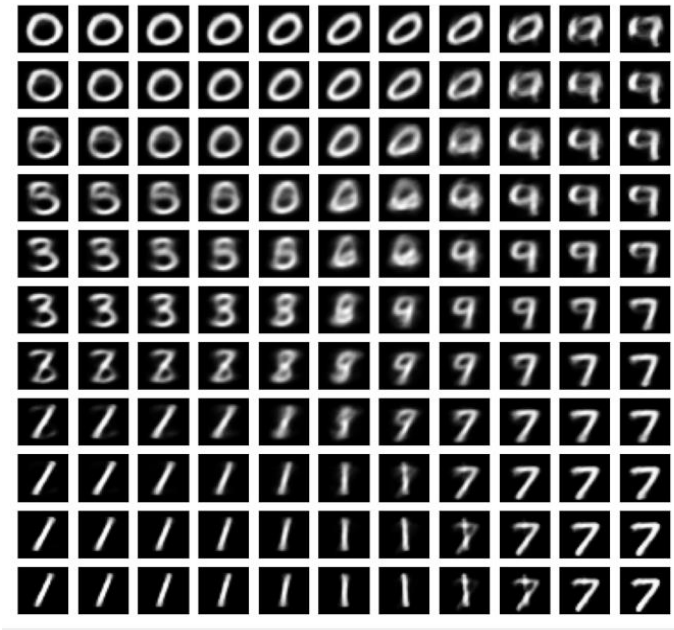

Рис. 3.9 – Семплінг точок з латентного простору

На даному зображенні бачимо, що не всі зображення задовільної якості, в деяких прикладах взагалі складно визначити цифру. Також можна помітити, що

такі цифри 2,4,5,6 взагалі майже не представлені. Можна зробити висновок, що наш латентний простір недостатньо описує вхідний сигнал. Для вирішення цієї проблеми, можна модернізувати архітектуру варіаційного автокодувальника.

3.1.2 Розширений варіаційний автокодувальник

Наша навчальна вибірка має мітки класів, тобто вказівник на те яку цифру має зображення, і цю мітку можна використати подаючи її на вхід кодера, та декодера. Дана модель має назву розширений варіаційний автокодувальник.

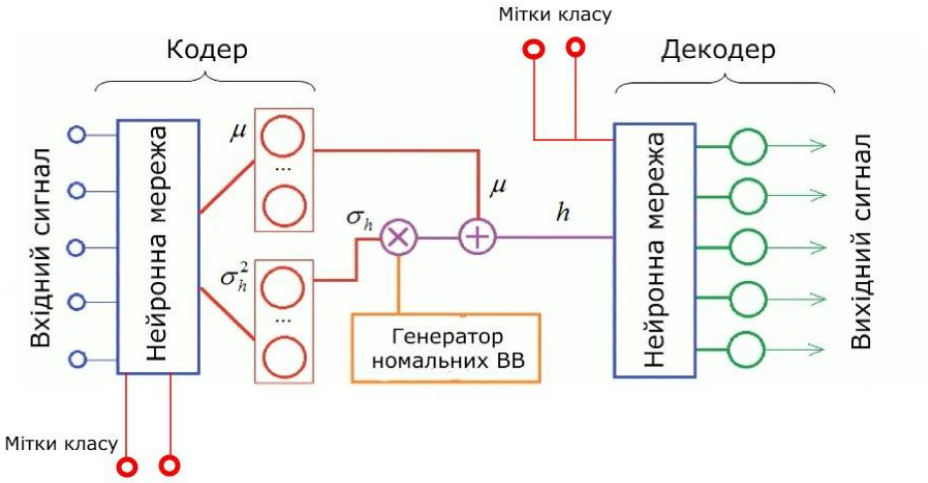

Рис. 3.10 – Розширений варіаційний автокодувальник

Таким чином наш декодер може інтерпретувати латентний простір, як певну цифру, яку ми вказуємо. Для прикладу передамо на вхід декодеру цифру 5, в результаті маємо отримати інтерпретацію різних точок з латентного простору у вигляді цифри 5. Тобто одна й та сама точка латентного простору інтерпретується по різному, в залежності від класу який ми передамо на вхід.

Рис. 3.11 – Генерація цифри 5

Це дозволяє також переносити стилі одного зображення на інше. Для прикладу ми на вхід кодера подаємо зображення цифри 5, та її мітку класу, а на вхід декодера передаємо іншу мітку класу, наприклад 7. Таким чином цифра 7 буде представлена у стилі написання цифри 5.

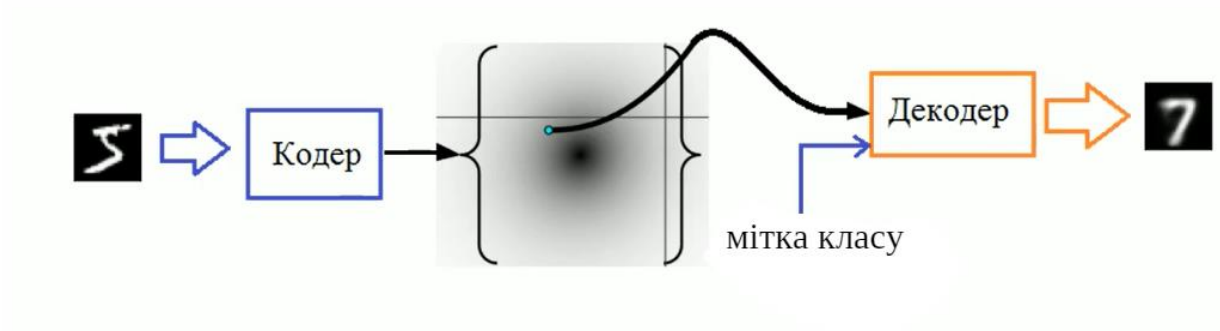

Рис. 3.12 – Схема перенесення стилю зображення

Проведемо даний експеремент для всіх класів, тобто візьмемо 10 зображень одного класу з навчального набору MNIST, та спробуємо відтворити стиль кожного зображення для інших класів.

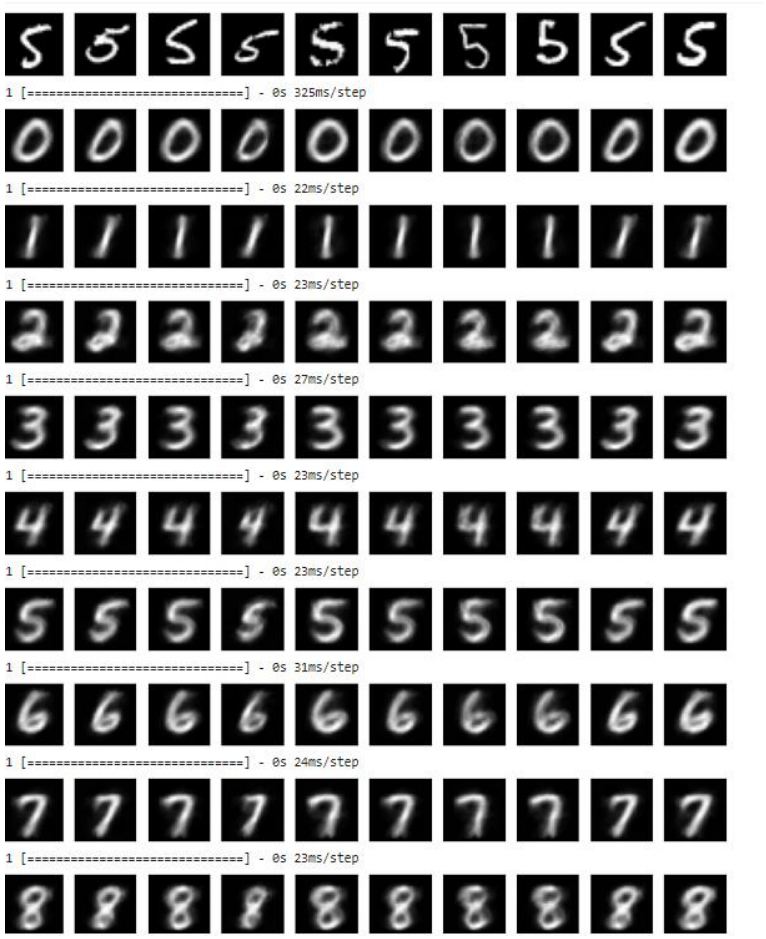

Рис. 3.13 – Перенесення стилю цифри 5 на зображення інших класів

В результаті можна зробити висновок, що кожний вектор латентного простору, описує деякі загальні характеристики кожного зображення, що може слугувати чудовим інструментом для генерації зображень, на основі навчальної вибірки.

3.2 Застосування розширеного варіаційного автокодувальника в задачі розпізнавання зображень

Варіаційний автокодувальник та згорткові нейронні мережі, це потужні методи глибокого навчання, об’єднання яких може забезпечити покращення точності та ефективності розпізнавання рукописних цифр на зображеннях. VAE є хорошим генеративним підходом, який навчається моделювати розподіл латентних змінних для вхідних даних.

У нашому випадку, VAE буде відповідати за взяття вхідного зображення та перетворення його у нові екземпляри, що дозволить збільшити розмір навчальної вибірки. CNN визначається своєю здатністю автоматичного вивчення просторових ієрархій ознак у зображеннях. Вона використовує згортки та пулінг для обробки зображень, що дозволяє визначати складні взаємозв'язки та розпізнавати об'єкти на зображеннях. Таким чином в цьому дослідженні об'єднуються VAE та CNN для визначення нового методу розпізнавання рукописних цифр на зображеннях. Вхідне зображення проходить через VAE, де використовуються кодувальний та декодувальний шари для аугментації навчальної вибріки, яка потім передається для навчання CNN, що виконує фінальне класифікаційне завдання.

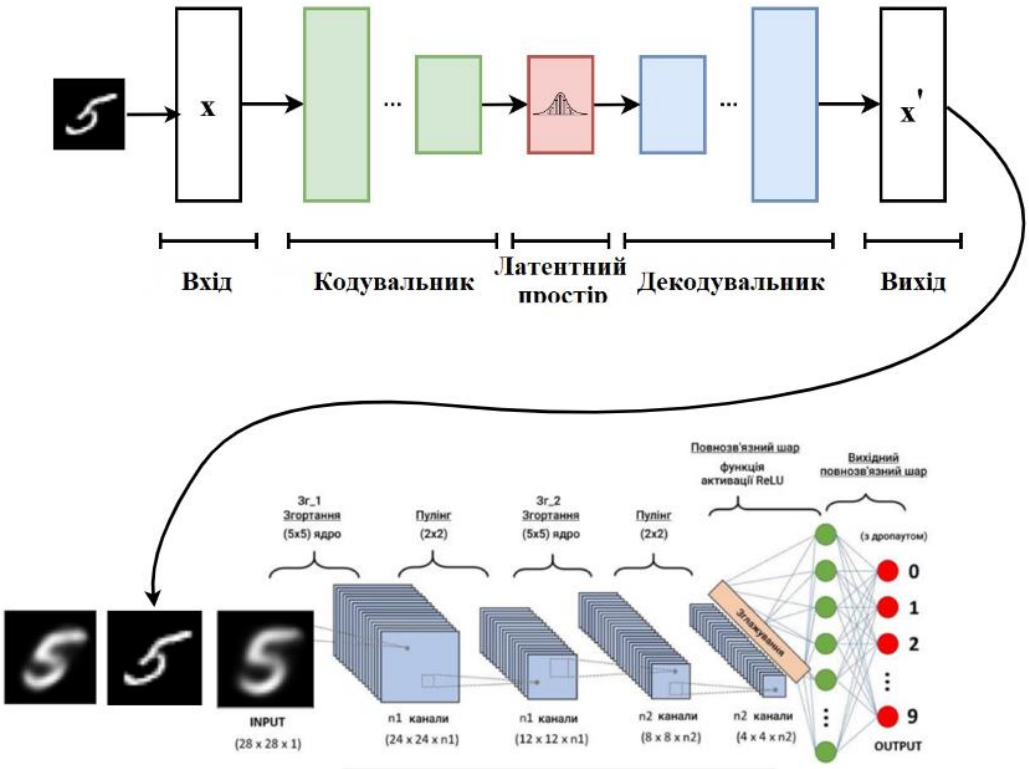

Рис. 3.14 – Поєднання VAE + CNN

Для детальнішого огляду алгоритму навчання моделі, проведемо більш конкретний опис:

1. Завантажуємо тренувальні та тестові дані MNIST, та випадковим чином обирається підмножина тренувальних даних з 200 елементів для емуляції ситуації з малою кількістю тренувальних даних. Проводимо нормалізацію даних.
2. Визначаємо архітектуру кодера для моделі (VAE). Кодер приймає на вході зображення та класову інформацію, та повертає середнє значення та логарифм дисперсії у латентному просторі. Вхідний шар для моделі, визначає форму вхідних даних у вигляді (28,28,1), тобто зображення розміром 28x28 пікселів з одним каналом (зображення в градаціях сірого). Далі визначаємо перший шар кодера, який має 256 нейронів та функцію активації ReLU. Другий шар кодера, приймає вихід попереднього шару на вхід, має 128

нейронів, та функцію активації ReLU. Повертається на вихід середнє значення та логарифм дисперсії у латентному просторі.

3. Вводимо шум у процес формування латентного простору, та формуємо його представлення.

4. Декодер приймає на вході представлення латентного простору та класову інформацію, і повертає відновлене зображення. Він складається з 2 повнозв'язних шарів із 128 та 256 нейронами та функцією активації ELU, та останнього шару з 28*28 (розмір зображення) кількістю нейронів та функцією активації sigmoid. Останній шар декодера видає відновлене зображення

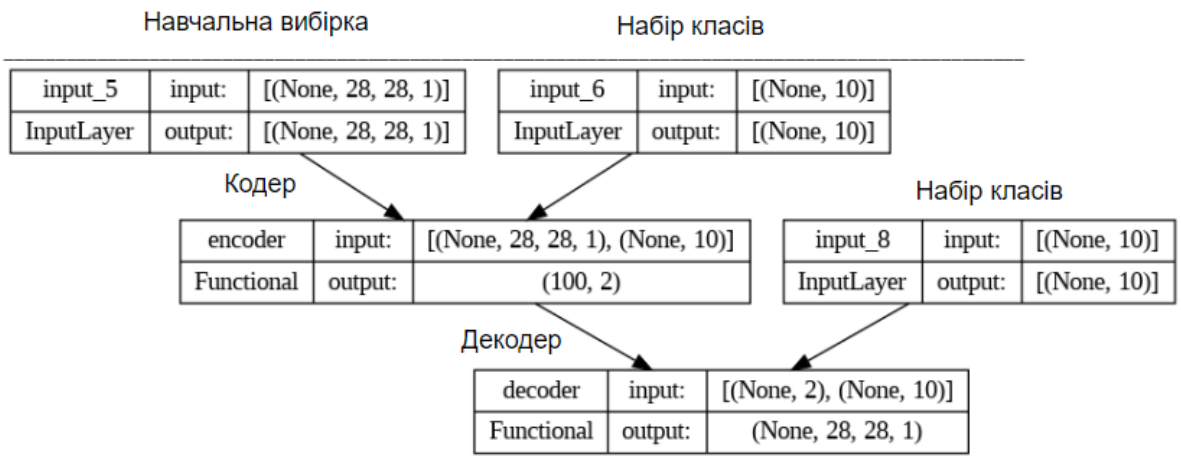

Рис. 3.15 – Архітектура VAE

5. Реалізовуємо функцію втрат яка включає у себе середній квадрат втрат та дивергенцію - KL.
6. Компілюємо та тренуємо модель VAE.

Після тренування можна згенерувати зображення на основі навчальної вибірки MNIST. На рисунку (3.16) для прикладу 15 оригінальних зображень, та 15 згенерованих зображень.

Рис. 3.16 – Згенеровані зображення

Таким чином ці зображення вже можна перемішати між собою, та передати на для навчання згорткової нейромережі. Розглянемо архітектура CNN, яка буде навчатися розпізнавати ці зображення.

Архітектура CNN

1. Завантажуємо та нормалізуємо набір згенерований за допомогою VAE. Додаємо один канал для зображень (для роботи зі згортковими шарами).
2. Створюємо модель CNN з наступною архітектурою:
    - згорткові шари з 32 та 64 нейронами, та функцією активації ReLU
    - пулінгові шари, з кроком 2, та розміром пікселя 2*2.
    - Перетворюємо вихідні дані у вектор для передачі повному з'єднувальному шару.
    - Повнозв’язний шар з 128 нейронами та функцією активації ReLU
    - Повнозв’язний шар з 10 нейронами, що відповідають кількості класів та функцією активації softmax для отримання ймовірностей принадлежності до кожного класу.

Layer (type)	Output Shape
conv2d_23 (Conv2D)	(None, 28, 28, 32)
max_pooling2d_14 (MaxPooling2D)	(None, 14, 14, 32)
conv2d_24 (Conv2D)	(None, 14, 14, 64)
max_pooling2d_15 (MaxPooling2D)	(None, 7, 7, 64)
flatten_11 (Flatten)	(None, 3136)
dense_26 (Dense)	(None, 128)
dense_27 (Dense)	(None, 10)

Рис. 3.17 – Архітектура CNN

Ця модель є класичною архітектурою згорткової нейронної мережі для класифікації зображень і включає згорткові шари для витягування рис об’єктів та повні з'єднувальні шари для класифікації зображень.

3.3 Оцінка ефективності

На основі побудованої моделі, можна провести оцінку ефективності розпізнавання зображень, шляхом порівняння результатів точності та втрат розпізнавання. Для порівняння будуть представлені результати розпізнавання за допомогою звичайної згроткової нейромережі з навчальною вибіркою 200 зображень, та за допомогою згорткової нейромережі, яка на вхід приймає аугментовану вибірку за домогою варіаційного автокодувальника з 400 зображень. Для справедливості порівнняння, налаштування CNN для розпізнавання абсолютно однакові, модель тренується 20 епох, з розміром батча 16.

```
10/10 [==============================] - 0s 26ms/step - loss: 0.0077 - accuracy: 1.0000 - val_loss: 0.6823 - val_accuracy: 0.8750
Epoch 14/20
10/10 [==============================] - 0s 27ms/step - loss: 0.0054 - accuracy: 1.0000 - val_loss: 0.7349 - val_accuracy: 0.8500
Epoch 15/20
10/10 [==============================] - 0s 26ms/step - loss: 0.0043 - accuracy: 1.0000 - val_loss: 0.7684 - val_accuracy: 0.8500
Epoch 16/20
10/10 [==============================] - 0s 26ms/step - loss: 0.0036 - accuracy: 1.0000 - val_loss: 0.7800 - val_accuracy: 0.8750
Epoch 17/20
10/10 [==============================] - 0s 28ms/step - loss: 0.0032 - accuracy: 1.0000 - val_loss: 0.7932 - val_accuracy: 0.8500
Epoch 18/20
10/10 [==============================] - 0s 28ms/step - loss: 0.0027 - accuracy: 1.0000 - val_loss: 0.7951 - val_accuracy: 0.8500
Epoch 19/20
10/10 [==============================] - 0s 26ms/step - loss: 0.0025 - accuracy: 1.0000 - val_loss: 0.8092 - val_accuracy: 0.8500
Epoch 20/20
10/10 [==============================] - 0s 26ms/step - loss: 0.0022 - accuracy: 1.0000 - val_loss: 0.8180 - val_accuracy: 0.8500
313/313 [==============================] - 5s 15ms/step - loss: 204.5889 - accuracy: 0.8241
313/313 [==============================] - 4s 13ms/step - loss: 204.5889 - accuracy: 0.8241
[204.5889434814453, 0.8241000175476074]
```

Рис. 3.18 - Результати навчання та розпізнавання CNN

```
Epoch 15/20
20/20 [==============================] - 0s 24ms/step - loss: 0.0156 - accuracy: 0.9937 - val_loss: 0.2026 - val_accuracy: 0.9250
Epoch 16/20
20/20 [==============================] - 1s 27ms/step - loss: 0.0045 - accuracy: 1.0000 - val_loss: 0.0851 - val_accuracy: 0.9375
Epoch 17/20
20/20 [==============================] - 0s 24ms/step - loss: 0.0017 - accuracy: 1.0000 - val_loss: 0.1620 - val_accuracy: 0.9250
Epoch 18/20
20/20 [==============================] - 1s 27ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.1680 - val_accuracy: 0.9500
Epoch 19/20
20/20 [==============================] - 0s 23ms/step - loss: 9.8296e-04 - accuracy: 1.0000 - val_loss: 0.1708 - val_accuracy: 0.9500
Epoch 20/20
20/20 [==============================] - 0s 25ms/step - loss: 8.3002e-04 - accuracy: 1.0000 - val_loss: 0.1682 - val_accuracy: 0.9500
313/313 [==============================] - 6s 17ms/step - loss: 131.7203 - accuracy: 0.8694
[131.7202606201172, 0.8694000244140625]
```

Рис. 3.19 - Результати розпізнавання CNN+VAE

За результатами навчання отримуємо наступні показники

Таблиця 3.1

Порівняння результатів

Метод	Кількість зображень на вході	Якість розпізнавання	Втрати
CNN	200	accuracy: 0.8241	loss: 204.588
CNN+VAE	200+200(згенеровані)	accuracy: 0.8694	loss: 131.720

В підсумку точність розпізнавання підвищено з 82% до 87%, втрати зменшилися з 204 до 131. Таким чином можна зробити висновок, що аугментація навчальної вибірки за допомогою варіаційного автокодувальника, дозволяє підвищити точність розпізнавання рукописних зображень в умовах недостатньої кількості навчальних даних приблизно на 5%

ВИСНОВКИ

1. Проведено огляд методів розпізнавання зображень, виділено основні переваги та недоліки. Розглянуто проблему використання згорткових мереж при недостатній кількості навчальних даних. В результаті запропоновано рішення за рахунок розширення навчальної вибірки.
2. Проаналізовано методи генерації зображень. Визначено що варіаційний автокодувальник може бути використаний для вирішення проблеми. Побудовано матиматичну модель та алгоритм варіаційного автокодувальника.
3. За допомогою розробленої моделі були згенеровані зображення, які намагаються відтворити зображення подане на вхід моделі. Виявлено що за допомогою розширеного варіаційного автокодувальника можна корегувати стиль генерованого зображення.Таким чином навчальну вибірку було збільшено в 2 рази.
4. Проведено аналіз ефективності розробленого методу для навчання згорткової мережі. Виявлено що використання розширеного варіаційного автокодувальника для розширення навчальної вибірки, дозволило підвищити точність розпізнавання зображень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Rajavelu, A., Musavi, M. T. & Shirvaikar, M. V. (1989). A neural network approach to character recognition. Neural Networks, (2(5)), 387–393. https://doi.org/10.1016/0893-6080(89)90023-3.

2. Bai, J., Chen, Zh., Feng, B. & Xu, Bo. Image character recognition using deep convolutional neural network learned from different languages. 2014 IEEE International Conference on Image Processing (ICIP 2014) (pp. 2560-2564). October 27-30, 2014, Paris, France: IEEE. DOI: 10.1109/ICIP.2014.7025518.

3. Maitra, D. S., Bhattacharya, U. & Parui, S. K. CNN based common approach to handwritten character recognition of multiple scripts. 3th International Conference on Document Analysis and Recognition (ICDAR), (pp. 1021-1025). August 23-26, 2015, Tunisia, Tuni. DOI:10.1109/ICDAR.2015.7333916.

4. Чичкарьов Є.А Зінченко О.В. Балалаєва О.Ю. Сергієнко А.В. Ковальов О.О Розпізнавання рукописних українських літер та цифр з використанням синтетичного набору даних та згорткових нейронних мереж, с.251 https://doi.org/10.36074/grail-of-science.23.12.2022.36

5. Lohvin A. O. TYPES OF GENERATIVE NEURAL NETWORKS. Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences. 2021. Vol. 1, no. 1. P. 102–109. URL: https://doi.org/10.32838/2663-5941/2021.1-1/17

6. ШЕРЕМЕТ, О. І.; САДОВОЙ, О. В. Метод опорних векторів (SVM). Математичне моделювання, 2013, 1: 13-17.

7. Paramud Y., Yarkun V. Алгоритмічно-програмні засоби розпізнавання рукописних символів на зображенні. Computer systems and network. 2017. Т. 1, № 1. С. 98–106. URL: https://doi.org/10.23939/csn2017.881.098 (дата звернення: 15.12.2023).

8. KATARIA, Aman; SINGH, M. D. A review of data classification using k-nearest neighbour algorithm. International Journal of Emerging Technology and Advanced Engineering, 2013, 3.6: 354-360.

9. Ayat N.E., Cheriet M., Suen C.Y.. Optimization of the SVM kernels using an empirical error minimization scheme. In Proc. of the International Workshop on Pattern Recognition with Support Vector Machine, Niagara Falls-Canada, August 2002, pp. 354-369.
10. Oliveira L.S., Sabourin R.: Support Vector Machines for Handwritten Numerical String Recognition. 9th International Workshop on Frontiers in Handwriting Recognition, October 2004.
11. Ahmad A.R., Viard-Gaudin C., Khalid M., Yusof R.: Online Handwriting Recognition using Support Vector Machine. Proceedings of the Second International Conference on Artificial Intelligence in Engineering & Technology, August 3-5 2004, pp. 250-256
12. Perez-Cortes J.C., Llobet R., Arlandis J.: Fast and Accurate Handwritten Character Recognition using Approximate Nearest Neighbours Search on Large Databases. https://prhlt.iti.upv.es/demos/demo_forms/Perez-cortes00b.pdf
13. Mico L., Oncina J.: Comparison of fast nearest neighbour classifier for handwritten character recogniton. Pattern Recognition Letters, 1999, 19 (3-4): 351-356.
14. Shouno H. Local Features in Image; Using HOG and SIFT. The Journal of the Institute of Image Information and Television Engineers. 2013. Vol. 67, no. 3. P. 256–258. URL: https://doi.org/10.3169/itej.67.256 (date of access: 15.12.2023).
15. Shorten C., Khoshgoftaar T. M. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data. 2019. Vol. 6, no. 1. URL: https://doi.org/10.1186/s40537-019-0197-0 (date of access: 15.12.2023).
16. Sohn, Kihyuk; Lee, Honglak; Yan, Xinchen Learning Structured Output Representation using Deep Conditional Generative Models URL: https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf
17. Bao, Jianmin; Chen, Dong; Wen, Fang; Li, Houqiang; Hua, Gang (2017). «CVAE-GAN: Fine-Grained Image Generation Through Asymmetric Training» URL: https://arxiv.org/abs/1703.10155

18. Fisher P. Descriptor matching with convolutional neural networks: a comparison to SIFT [Електронний ресурс] / P. Fisher, A. Dosovitskiy // arXiv preprint arXiv:1405.5769. – 2014. – Режим доступу до ресурсу: https://arxiv.org/abs/1405.5769.
19. САМОЛЮК, Т. А. Нейромережі GAN в створенні нових моделей. Комп'ютерні засоби, мережі та системи, 2019.
20. MELNYK, V.; MELNYK, K.; KOPTIUK, Ju. Дослідження методів розпізнавання зображень на основі нейронних мереж. COMPUTER-INTEGRATED TECHNOLOGIES: EDUCATION, SCIENCE, PRODUCTION, 2019, 35: 161-165.
21. Котул О.Ю. Використання нейромереж для генерації зображень // Всеукраїнська науково-технічна конференція «Тхнологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і Світу». – Київ: ДУІКТ, 2023. с.185-186
22. Котул О.Ю. Аналіз проблем навчання нейронних мереж // XVII Міжнародна науково-практична конференція «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» – Дніпро: УДУНТ, 2023 с.117

ДЕМОНСТРАЦІЙНІ МАТЕРАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

«Розробка методу розпізнавання рукописних зображень на основі варіаційного автокодувальника»

Виконав: студент групи ПДМ-61 Котул Олександр Юрійович

Керівник: к.т.н., доц., ІПЗ Негоденко Олена Василівна

Київ - 2024

МЕТА, ОБ'ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: покращення точності розпізнавання рукописних зображень за рахунок варіаційного автокодувальника

Об'єкт дослідження: процес розпізнавання рукописних зображень

Предмет дослідження: метод для розпізнавання зображень на основі варіаційного автокодувальника

АКТУАЛЬНІСТЬ РОБОТИ

Методи розпізнавання зображень

Метод	Переваги	Недоліки
Метод k-найближчих сусідів	Алгоритм легко зрозуміти і візуалізувати. Він не вимагає складних математичних розрахунків чи параметрів. Не потребує навчання, оскільки модель зберігає весь тренувальний набір даних, і прогнозується на основі схожості між тестовим прикладом і найближчими сусідами.	При великому обсязі даних обчислення відстаней між всіма точками може займати дуже багато часу, що робить k-NN менш ефективним для великих наборів даних. Також у високорозмірних просторах всі точки стають однаково віддаленими, що може знизити ефективність методу.
Метод опорних векторів (SVM)	Добре справляється з високорозмірними просторами, що робить його ефективним для задач, де кількість вхідних ознак велика, наприклад, у задачах обробки зображень чи тексту	Схильний до перенавчання, особливо коли дані несбалансовані, та може стати обчислювально вимогливим при великому обсязі даних, оскільки час обчислень зростає квадратично з розміром набору даних.
Згорткові нейронні мережі (CNN)	CNN добре враховує просторові характеристики вхідних даних, що робить його ефективним у завданнях обробки зображень, та може автоматично вивчати та виділяти важливі ознаки зображень без необхідності ручного визначення характеристик.	Для успішного навчання глибоких CNN часто потрібна велика кількість даних, що може бути викликом для задач з обмеженим обсягом даних. При використанні невеликої кількості даних CNN може стати схильним до перенавчання.

Порівняння результатів розпізнавання рукописних зображень

Метод	Показники точності при розмірі вибірки 200 елементів	Оптимальний розмір вибірки	Точність розпізнавання зображень
Метод k-найближчих сусідів	±90%	1000	95%
Метод опорних векторів (SVM)	±77%	4500	93%
Згорткові нейронні мережі (CNN)	±80%	4000	98%

Методи генерації зображень

Метод	Переваги	Недоліки
Генеративно-суперечлива мережа (GAN)	Створює різкі зразки зображень. Може генерувати різноманітні зображення, що робить його ефективним для творчих завдань, таких як генерація мистецтва або дизайну.	Складніше оптимізувати через нестабільну динаміку тренування. GAN може вивчити занадто конкретні особливості тренувальних даних і стати непридатним для генерації нових, реалістичних зображень.
Авторегресивні мережі	Мають просту математичну формулу та є легкими для інтерпретації, що робить їх привабливими для застосування в контексті, де важлива зрозумілість результатів.	Неефективні під час вибірки і не можуть легко забезпечити низькорозмірні функції
Варіаційний автокодер (VAE)	Одночасно виконує генерацію і логічний висновок із моделюванням прихованих змінних.Є можливість використовувати різні архітектури та варіанти модифікацій, дозволяє адаптувати VAE до конкретних завдань.	Зразки зображень, згенеровані VAE, мають тенденцію бути трохи розмитими

Модель автокодувальника

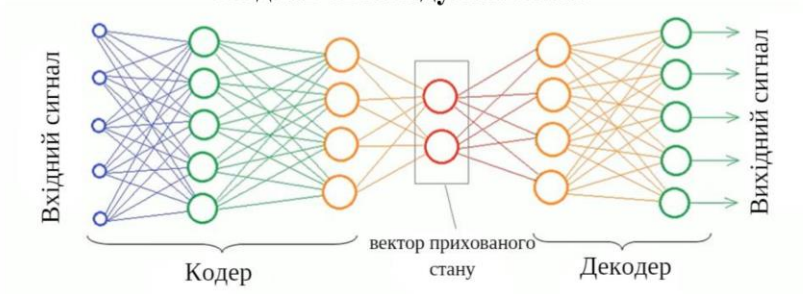

Рис. 6.1 Алгоритм автокодувальника

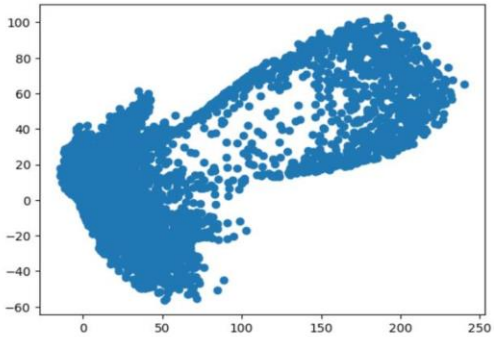

Рис. 6.2 Розподіл автокодувальника в латентному просторі

Модель варіаційного автокодувальника

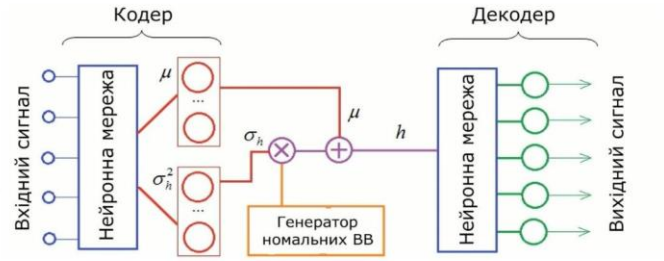

Рис. 7.1 Алгоритм варіаційного автокодувальника

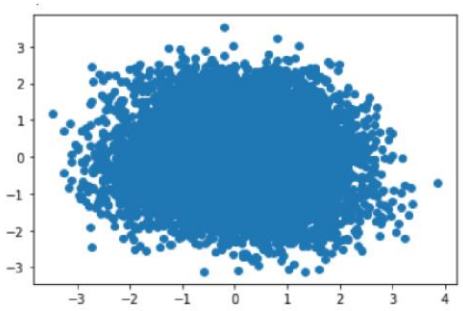

Рис. 7.2 Розподіл варіаційного автокодувальника в латентному просторі

Модель розширеного варіаційного автокодувальника

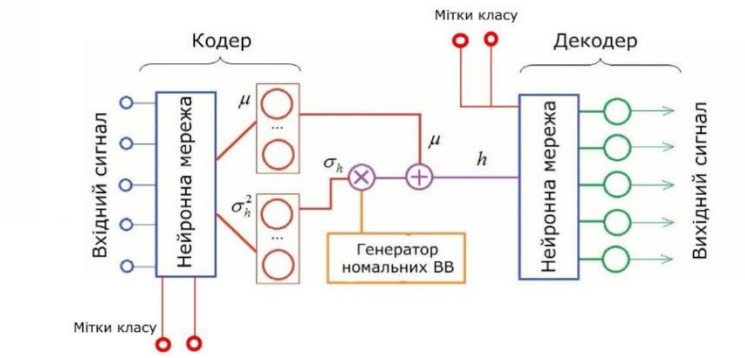

Рис. 8.1 Алгоритм розширеного варіаційного автокодувальника

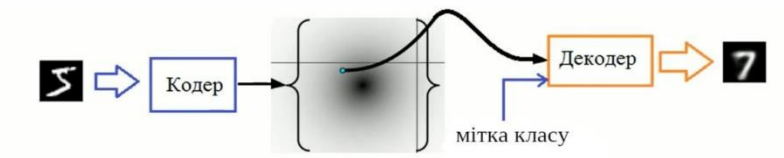

Рис. 8.2 Алгоритм передачі стилю з одного зображення на інше

Модель розробленого методу

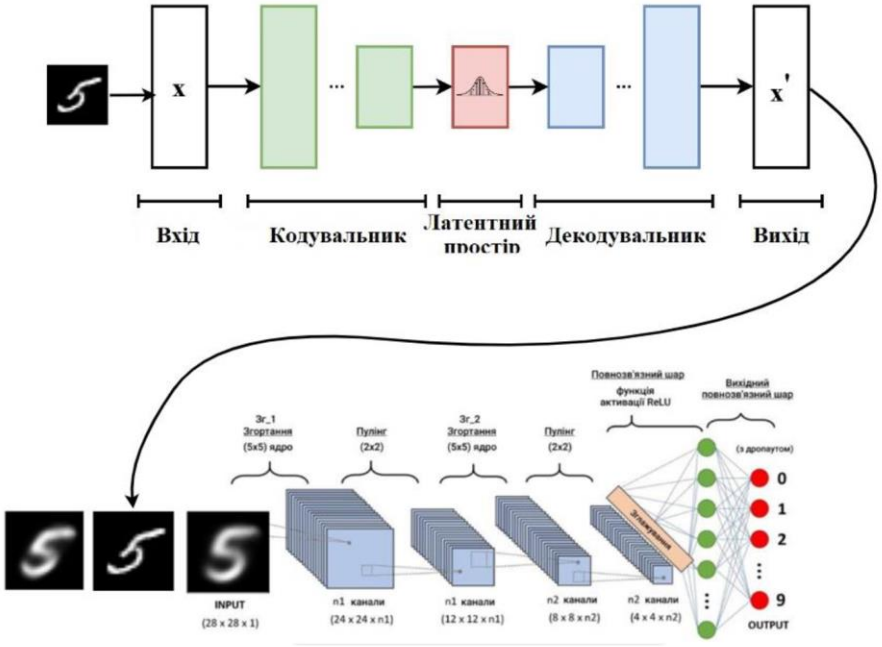

Рисунок 10.1 Варіаційний кодувальник + згорткова нейромережа

ПРАКТИЧНИЙ РЕЗУЛЬТАТ

Рис.9.1 Інтерпретація різних точок з латентного простору у вигляді цифри 5

Рис.9.2 Перенесення стилів на інші класи

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

Котул О.Ю. Використання нейромереж для генерації зображень // Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і Світу». – Київ: ДУІКТ, 2023. с.185-186

Котул О.Ю. Аналіз проблем навчання нейронних мереж // XVII Міжнародна науково-практична конференція «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» – Дніпро: УДУНТ, 2023 с.117

Стаття:

Котул О.Ю. Аналіз методів для розпізнавання та синтез зображень для розширення навчальної вибірки // Зв'язок 2023 (прийнято до друку)

ДЯКУЮ ЗА УВАГУ!