

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»

ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису  
УДК 51-78

До захисту допущено  
Завідувач кафедри ММСА  
О.Л.Тимошук  
«\_\_\_» \_\_\_\_\_ 2021 р.

**Магістерська дисертація**

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз  
на тему: «Вирішення транспортної задачі методами машинного  
навчання»

Виконала:  
студентка II курсу, групи КА-01мп  
Гайдук Ірина Вадимівна \_\_\_\_\_

Керівник: доцент кафедри ММСА,  
к.ф.-м.н., доц., Яковлева А.П. \_\_\_\_\_

Рецензент:  
в. о. завідувача кафедри системного проектування  
КПІ ім. Ігоря Сікорського,  
д.т.н., проф., Мухін В.Є. \_\_\_\_\_

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів  
без відповідних посилань  
Студент \_\_\_\_\_

Київ  
2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»

ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)  
Спеціальність — 124 «Системний аналіз»

ЗАТВЕРДЖУЮ

Завідувач кафедри ММСА

О. Л. Тимошук

«\_\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**

на магістерську дисертацію студентці Гайдук Ірині Вадимівні

**1. Тема дисертації:** «Вирішення транспортної задачі методами машинного навчання», науковий керівник дисертації Яковлева Алла Петрівна, к.ф.-м.н., доц., затверджені наказом по університету від “02” листопада 2021 № 3651-с.

**2. Термін подання студентом дисертації:** 10 грудня 2021 р.

**3. Об’єкт дослідження:** транспортна задача, задача оптимального транспортування.

**4. Предмет дослідження:** методи машинного навчання, зокрема генеративна змагальна нейронна мережа.

**5. Перелік завдань, які потрібно розробити:**

1) провести аналіз актуальності задачі та визначення сфер використання;

2) провести аналіз існуючих методів вирішення задачі оптимального транспортування;

3) побудувати нейронну мережу для знаходження оптимального відображення;

4) застосувати побудовану модель для трьох різних наборів даних;

5) розробити стартап-проект виведення на ринок результатів дослідження;

6) розробити концептуальні висновки за результатами наукового дослідження;

**6. Орієнтовний перелік графічного (ілюстративного) матеріалу:**

**7. Дата видачі завдання:** 01 вересня 2021 р.

**Календарний план**

| № з/п | Назва етапів виконання магістерської дисертації                                                                   | Термін виконання етапів магістерської дисертації |
|-------|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| 1.    | Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі, завдань, практичної значущості результатів |                                                  |
| 2.    | Перший розділ. Формулювання транспортної задачі та її розповсюдженість                                            |                                                  |
| 3.    | Другий розділ. Дослідження існуючих методів                                                                       |                                                  |
| 4.    | Третій розділ. Побудування та навчання нейронної мережі. Аналіз результатів                                       |                                                  |
| 5.    | Четвертий розділ. Стартап-проект                                                                                  |                                                  |
| 6.    | Концептуальні висновки. Перспективи розвитку отриманих рішень                                                     |                                                  |

Студентка

І.В. Гайдук

Науковий керівник дисертації

А.П. Яковлева

## РЕФЕРАТ

Магістерська дисертація: 87 с., 27 рисунків, 24 таблиці, 21 джерело.

В роботі розглянута класична задача оптимального транспортування. Проведено дослідження відомих методів її вирішення, їх переваги та недоліки, необхідні умови існування оптимального розв'язку. Окрім цього, був запропонований машинний метод вирішення задачі з побудовою та навчанням моделі на основі генеративної нейронної мережі.

В роботі було розглянуто загальні відомості про методи вирішення задачі оптимального транспортування при її незбалансованості та масштабованості. Було виконано аналіз результатів трьох різних типів задач, вирішених методом машинного навчання.

Об'єктом дослідження є класична задача оптимального транспортування у трьох різних видах.

Предметом дослідження є методи машинного навчання, зокрема генеративна змагальна нейронна мережа.

МАШИННЕ НАВЧАННЯ, НЕЗБАЛАНСОВАНА  
ТРАНСПОРТНА ЗАДАЧА, ЗАДАЧА ОПТИМАЛЬНОГО  
ТРАНСПОРТУВАННЯ, НЕЙРОННІ МЕРЕЖІ, ГЕНЕРАТИВНА  
ЗМАГАЛЬНА НЕЙРОННА МЕРЕЖА.

## ABSTRACT

Master's thesis: 87 pages, 27 figures, 24 tables, 21 sources.

Theme: The classical problem of optimal transportation. The conducted research solves it by known methods, their advantages and disadvantages, the necessary conditions for the existence of an optimal solution. This was a proposed machine method for solving problems with the construction and model of learning based on a generative neural network.

The paper considered general information on the method of solving the problem of optimal transportation with its unbalance and scalability. The results of three different types of problems solved by the machine learning method were analyzed.

The subject of the study is the classical problem of optimal transportation in three different types.

The subject of research is the methods of machine learning, in particular the generative competitive neural network.

MACHINE LEARNING, UNBALANCED OPTIMAL  
TRANSPORTATION, NEURAL NETWORKS, GENERATIVE  
COMPETITIVE NEURAL NETWORK

## ЗМІСТ

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| ВСТУП                                                                                      | 8  |
| РОЗДІЛ 1 ФОРМУЛЮВАННЯ ТРАНСПОРТНОЇ ЗАДАЧІ ТА ЇЇ РОЗПОВСЮДЖЕНІСТЬ                           | 10 |
| 1.1. Вступ                                                                                 | 10 |
| 1.2. Способи використання транспортної задачі для вирішення задач із різних галузей        | 11 |
| 1.3. Формулювання задачі оптимального транспортування                                      | 14 |
| 1.4. Відстань Васерштайна                                                                  | 19 |
| 1.5. Незбалансовані задачі оптимального транспортування                                    | 22 |
| 1.6. Аналіз Прокруста                                                                      | 25 |
| 1.7. Висновки до розділу                                                                   | 26 |
| РОЗДІЛ 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ                                                      | 28 |
| 2.1. Вступ                                                                                 | 28 |
| 2.2. Лінійне програмування                                                                 | 28 |
| 2.3. АНА метод                                                                             | 32 |
| 2.4. Регульоване оптимальне транспортування. Алгоритм масштабування Сінкхорна та Грінхорна | 35 |
| 2.5. Машинне навчання для оптимізації функцій                                              | 41 |
| 2.6. Використання дерев для незбалансованих задач                                          | 43 |
| 2.7. Висновки до розділу                                                                   | 46 |
| РОЗДІЛ 3. ПОБУДОВА ТА НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ. АНАЛІЗ РЕЗУЛЬТАТІВ                        | 47 |
| 3.1. Розгляд генеративної змагальної мережі та її адаптація. Архітектура мережі            | 47 |
| 3.2. Опис програмного забезпечення                                                         | 49 |
| 3.3. Набори вхідних даних                                                                  | 50 |
| 3.4. Аналіз результатів                                                                    | 54 |
| 3.5. Висновки до розділу                                                                   | 60 |
| РОЗДІЛ 4 РОЗРОБКА СТАРТАП ПРОЕКТУ                                                          | 62 |
| 4.1. Опис ідеї проекту                                                                     | 62 |

|                                                          |    |
|----------------------------------------------------------|----|
|                                                          | 7  |
| 4.2. Технологічний аудит ідеї проекту                    | 64 |
| 4.3. Аналіз ринкових можливостей запуску стартап-проекту | 66 |
| 4.4. Розроблення ринкової стратегії проекту              | 73 |
| 4.5. Розроблення маркетингової програми стартап-проекту  | 75 |
| 4.6. Висновок до розділу                                 | 77 |
| ПЕРЕЛІК ПОСИЛАНЬ                                         | 79 |
| ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ                              | 82 |

## ВСТУП

Сфери використання транспортної задачі є неймовірно різноманітними. В останні 12 років методи з використанням оптимального відображення отримали 2 медалі Філдса — нагороді, що видається 2-4 математикам на Міжнародному конгресі математиків.

Використання задачі оптимального відображення у машинному навчанні стали однією з оптимальних тем, дозволяючи досліджувати можливість використання трансферного навчання, знаходження мірі схожості зображень, заміна кольорової гами, і, звичайно, вирішення логістичної задачі.

Машинне навчання — прогресивний напрямок у вирішенні задач, де необхідно знаходити зв'язок між даними, зокрема задачі класифікації, генерації, регресії, прогнозування тощо. Для задачі оптимального транспортування є необхідним знаходження такого відображення, що буде оптимально переводити один розподіл у інший. З цією задачею є оптимальним використання генеративної змагальної мережі, що здатна навчатися без учителя та без початкової обробки даних.

Метою цієї роботи є дослідження використання методів машинного навчання для вирішення транспортної задачі. Необхідною умовою є можливість побудови нейронної мережі на різних типах задач для різних сфер використання, розв'язання незбалансованої та масштабованої задачі.

Об'єктом дослідження магістерської дисертації є класична задача оптимального транспортування у трьох різних видах.

Предметом дослідження є методи машинного навчання, зокрема генеративна змагальна нейронна мережа.

В ході роботи було поставлено та вирішено наступні задачі:

1. Дослідження транспортної задачі, її видів та типів, актуальність та сфери використання.

2. Дослідження існуючих методів вирішення транспортної задачі, виявлення їх недоліків та умов існування оптимального рішення .
3. Проектування та розробка архітектури генеративної змагальної нейронної мережі.
4. Аналіз результатів генерації оптимального відображення на трьох різних наборах даних.

## РОЗДІЛ 1 ФОРМУЛЮВАННЯ ТРАНСПОРТНОЇ ЗАДАЧІ ТА ЇЇ РОЗПОВСЮДЖЕНІСТЬ

### 1.1. Вступ

Одним з ключових факторів в ціноутворенні товару є логістична складова. До тих пір, поки не винайшли телепортацію, бізнес потребує наявності оптимального, з точки зору швидкості та вартості, способу перевезення. З вирішенням задачі про оптимальне розподілення ресурсів кожен день “стикаються” як власники маленьких кав’ярень, логісти на великих підприємствах, так і кінематографісти при заміні обличчя одного актора на інше. Адже в епоху процвітання машинного навчання, задачу транспортування/перенесення даних з одного стану в інший при певних умовах з мінімальними затратами на перевезення можна розглядати не лише як логістичну задачу, а накласти математичну модель на технічні проблеми реконструкції форми в комп’ютерній графіці, автоматичного перекладу при використанні нейромереж, зміни кольору на картинці, трансферне навчання тощо.

Для транспортної задачі як у вигляді задачі лінійного програмування, так і у вигляді задачі перетворення розподілів за найменших зусиль, існує безліч способів вирішення. Від найбільш примітивного методу перебору, знаходження відстані Васерштайна (англ. Wasserstein Distance) до використання нейронної мережі або дерева рішень. Кожен метод має свої переваги та недоліки, які визначаються через об’єм даних, зону використання, кількість та складність умов, область знаходження рішень. В цій роботі розглянуто методи машинного навчання, їх переваги та недоліки на декількох типах задач з різними вхідними параметрами.

Актуальність даної теми аргументовано розміром зони використання, потенційними можливостями використання методу в різноманітних галузях знань.

## 1.2. Способи використання транспортної задачі для вирішення задач із різних галузей

В зв'язку з відносною простотою формулювання задачі знаходження оптимального шляху, її модель можна використати в багатьох сферах. В цьому підрозділі будуть розглянуті певні приклади використання теорії оптимального транспортування, що доведе актуальність швидкого знаходження рішення для таких типів задач.

24 вересня 2020-го року, компанія Microsoft опублікувала на своєму офіційному сайті статтю про “Знаходження міри схожості двох наборів даних з використанням теорії оптимального транспортування” [1]. В цьому дослідженні було розглянута задача знаходження міри схожості двох розмічених наборів даних для виявлення можливості використання вже навченої на першому наборі моделі на другому наборі або можливість аугментації даних з одного набору даними іншого (рисунок 1.1).

Ще однією задачею, з якою справляється оптимальне транспортування є передача ознак одного об'єкту на інший. Зміна кольорової гами одного малюнка, кольоровою гамою іншого як на рисунку 1.2 [3], інтерполяція форми як на рисунку 1.3 [2], рис обличчя людини, заміна обличчя на фото та відео, пошук найбільш схожого зображення.

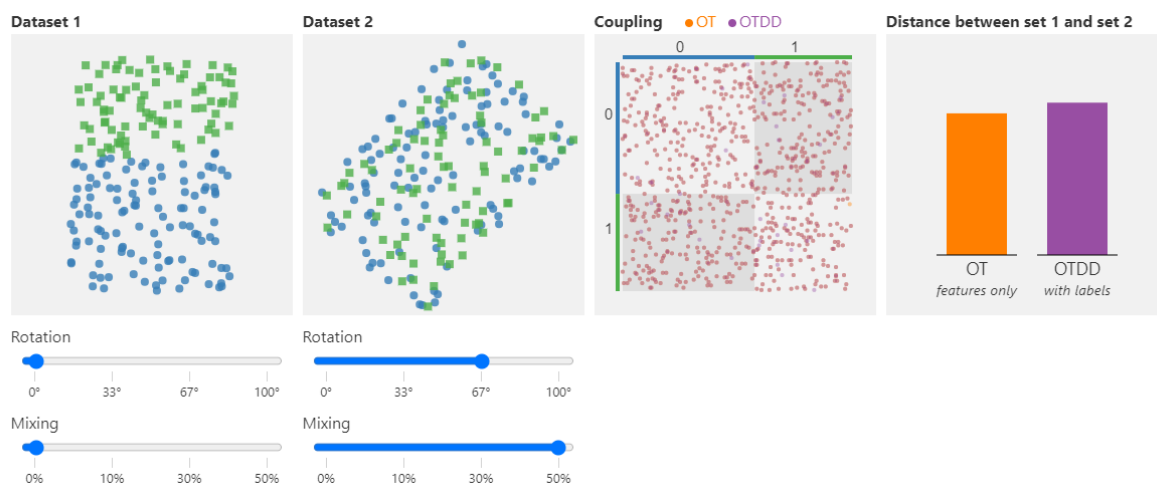


Рисунок 1.1 - Знаходження відстані Васерштайна між двома наборами даних.

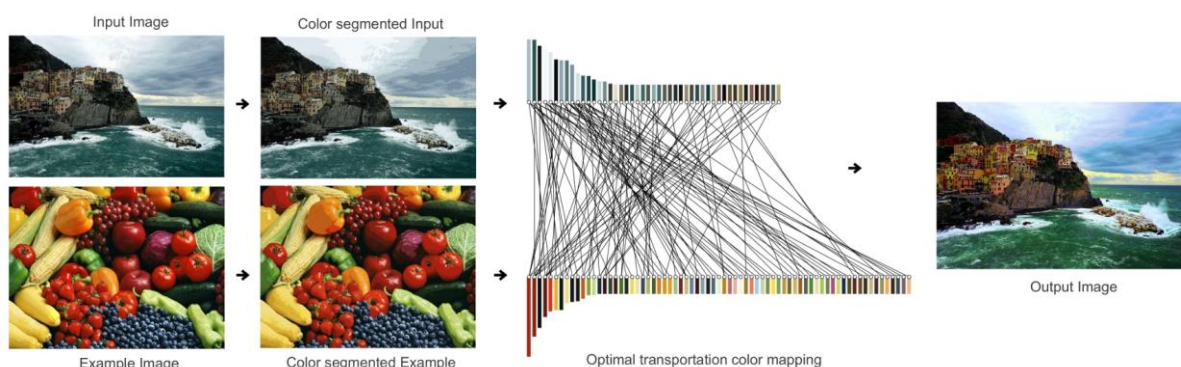


Рисунок 1.2 - Перенесення кольору з картинки

Методи пошуку фрагменту картинки за допомогою оптимального транспортування допомагають у медицині при діагностуванні захворювань, при знаходженні схожих картинок в великому наборі даних.

Наступний, і не менш важливий кластер задач - логістичні задачі. Теорія транспортної задачі зародилась саме на проблемі перевезення товарів з заводів до магазинів. В 1930 році О.М. Толстой опублікував першу в цій сфері задач статтю “Методи визначення мінімального кілометражу при транспортуванні вантажів в космосі” [4].

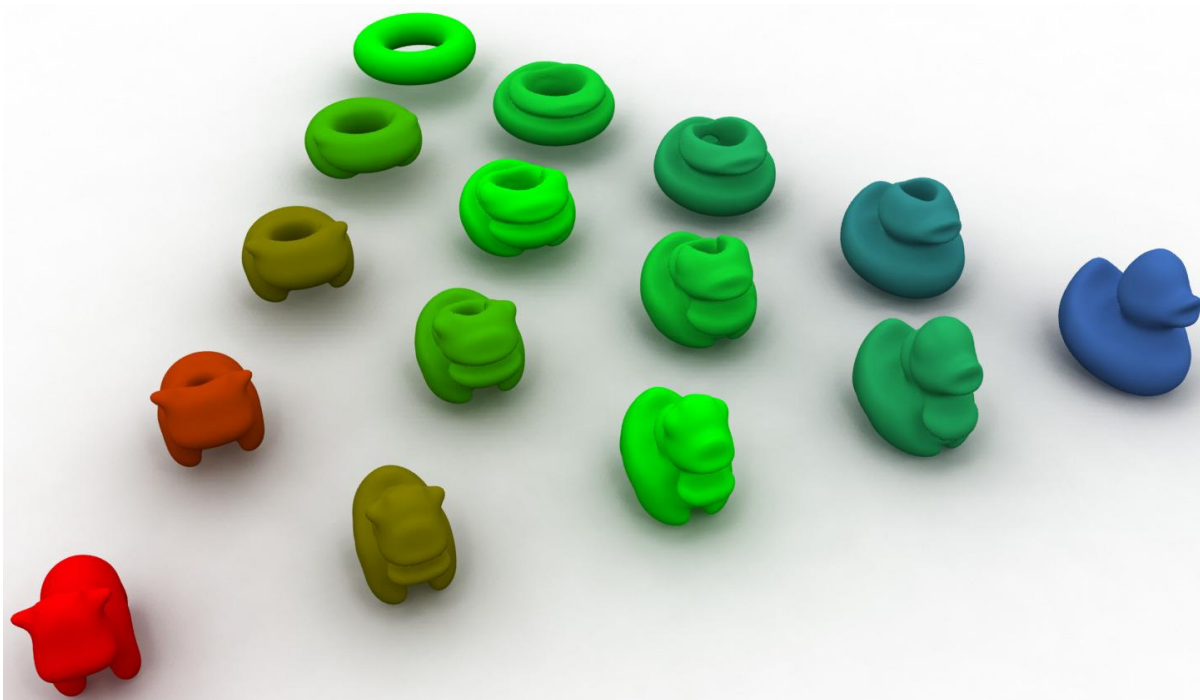


Рисунок 1.3 - Інтерполяція форми методом оптимального транспортування

Наразі логістичні задачі набули більшої важкості через різноманіття товарів, способів перевезення, умов перевезення та збільшення і ускладнення трафіку.

Задачу розподілення таксистів по замовленням також можна вирішити знаходженням мінімальної дистанції до кожного працівника.

При роботі з великими сховищами даних також використовують теорію оптимального транспортування або його модифікації. Імітація та інтерполяція відсутніх даних [5], допомога при трансферному навчанні при переході з одного домену до іншого, для класифікації даних, знаходження регресійної залежності.

Для підприємств задачі розподілу працівників між ділянками, розташування робочих місць в цеху, використання апаратів для різних видів робіт можуть бути вирішені при формулюванні відповідної моделі транспортної задачі.

Робота з текстом, а особливо з автоматичним перекладом, також є задачею, яку можна розв'язати, використовуючи оптимальне

транспортування. Для двох словників даних ведеться пошук відстані Громова-Васерштейна між розподілами їх слів [6]. Знайшовши цю міру, можна дослідити схожість мов між собою як на рисунку 1.4.

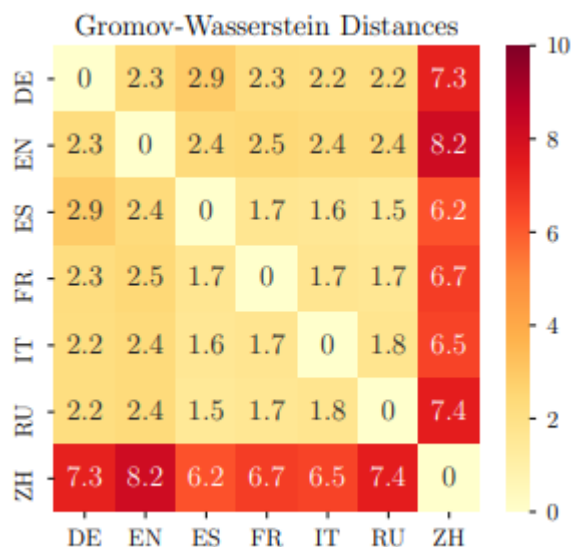


Рисунок 1.4 - Схожість розподілів мов

Цікавим висновком з вищевказаної роботи була схожість російської та іспанської мов, що на перший погляд є вкрай не очевидною, але що можна помітити придивившись до синтаксичної складової двох мов.

Великим плюсом у використанні теорії оптимального транспорту є можливість використання немаркованих даних тобто навчання без учителя.

### 1.3. Формулювання задачі оптимального транспортування

Класична транспортна задача була сформована як рішення транспортування товарів із виробництва у пункти споживання при

виконанні певних умов по об'єму товарів, попиту при досягненні мінімальної ціни.

Задача оптимального транспортування, або задача Монжа являє собою знаходження відображення з одного простору в інший [7]. Дано два ймовірнісних простори  $(X, A, \mu)$ ,  $(Y, B, \nu)$  та невід'ємна вимірна функція  $h(x, y)$  на  $X \times Y$ , що також називається функцією вартості. Задачею Монжа є задача знаходження відображення  $T: X \rightarrow Y$ , вимірне відносно пари  $(A, B)$ . Відображення  $T$  переводить міру  $\mu$  і міру  $\nu$ . Фіналізуючи вищевикладені умови, задача оптимального транспортування полягає у знаходженні такого відображення  $T$ , що:

$$\begin{aligned} M(\mu, \nu, T) &= \int_X h(x, T(x)) \mu(dx) \rightarrow \min, \\ T^{-1}(C) &\in A, \forall C \in B, \\ \nu \circ T^{-1}(C) &= \mu(T^{-1}(C)), \forall C \in B \end{aligned} \quad (1.1)$$

Задача Монжа не завжди має рішення навіть для дуже простих просторів, мір і функцій. Тим більше дивно, що при досить широких припущеннях ця задача має наближені рішення до задачі Канторовича.

В задачі Канторовича [8] замість пошуку відображення  $T$  ціллю стає пошук ймовірнісної міри  $\hat{b}$  на  $(X \times Y, A \otimes B)$ , що її проекція на  $X$  є  $\mu$ , а на  $Y$  —  $\nu$ . При мірі  $\hat{b}$  досягається мінімум виразу:

$$K(\mu, \nu, \hat{b}) = \min \left( \int_{X \times Y} h(x, y) \hat{b}(dx dy) \right), \forall \hat{b} \in \Pi(\mu, \nu) \quad (1.2)$$

Задача Канторовича, на відмінність від задачі Моржа, є лінійною. В залежності від виду функції  $h(x, y)$ , функціонал  $\hat{b} \mapsto K(\mu, \nu, \hat{b})$  також буде неперервним. При компактності просторів  $X$  та  $Y$ ,  $\Pi(\mu, \nu)$  також стає компактом.

При певних умовах, є корисним використання формулювання двоїстої задачі. Вводиться множина пар  $(\varphi, \psi)$  такі, що  $\varphi: X \rightarrow R^1$ ,  $\psi: Y \rightarrow R^1$ , вимірні відносно  $A$  та  $B$ , що задовольняють нерівності:

$$\varphi(x) + \psi(y) \leq h(x, y), \quad x \in X, y \in Y \quad (1.3)$$

І для множини пар вводиться двоїста задача знаходження величини:

$$J(\mu, \nu) = \sup \left( \int_X \varphi d\mu + \int_Y \psi d\nu \right) \quad (1.4)$$

Має місце оцінка:

$$J(\mu, \nu) \leq K(\mu, \nu) \leq M(\mu, \nu) \quad (1.5)$$

Існування рішення задачі Канторовича можливе при належність мір  $\mu$  та  $\nu$  до Радонових та цілком регулярному просторі (в ньому для  $\forall$  замкненої множини  $B$  та точки  $y$  поза множиною існує неперервна функція  $f(x)$  така, що  $(f(x) = 0, \forall x \in B)$  та  $(f(y) = 1, y \notin B)$ ) та наявність властивості напівнеперервності знизу для функції  $h(x)$ . Міра є Радоновою коли вона визначена на  $\sigma$ -алгебрі борелівських множин на Гаусдорфовому топологічному просторі та є локально кінцевою (для кожної точки існує окіл, міра на якому є кінцевою) та внутрішньо-регулярною (міра на борелівській множині співпадає з супремумом міри компактних підмножин цієї борелівської множини). Простір є топологічним, коли для кожної множини з цього простору визначені операції перетину, об'єднання, пуста множина належить цьому простору. Гаусдорфів топологічний простір задовольняє аксіомі віддільності, тобто для будь-яких двох точок існують околиці, що не

перетинаються. Борелівська сігма-алгебра містить всі відкриті підмножини топологічного простору.

Дискретна транспортна задача має вигляд саме задачі лінійного програмування. При цьому відображення  $T$  перетворюється в матрицю ціни  $C$  перевезення з дискретної множини  $X$  в дискретну множину  $Y$ . Множини  $X \cap Y = \emptyset$ . Інтеграл по дискретній множині значень дорівнює сумі. Тож класична дискретна задача оптимального транспортування має вигляд:

$$\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \rightarrow \min$$

При умовах:

$$\sum_{j=1}^m x_{ij} \leq a_i, a_i \in X \quad (1.6)$$

$$\sum_{i=1}^n x_{ij} = b_j, b_j \in Y$$

$$x_{ij} \geq 0$$

Коли загальний об'єм пропозицій дорівнює сумі попиту, то задача називається замкненою, або збалансованою. Часто замість матриці витрат на транспортування використовується час на перевезення, лінійна оболонка з факторів, що впливають на попит, у більш складних випадках нелінійні, стохастичні функції

Для дискретної задачі існує декілька узагальнень, в тому числі:

- мережева транспортна задача, при якій не існує поділу зв'язку між точками виробництва та точками продукції. Граф, що будується на основі цієї задачі може мати будь-які зв'язки між точками, проте точка виробництва задається позитивним числом, точка збуту — негативним;

- задача з обмеженням пропускної здатності, де дуги графу задаються з певним позитивним значенням, що і означає її пропускну здатність;
- багатопродуктова задача, що найбільш наближена до реальних задач оптимального транспортування.

У деяких випадках, особливо при наявності великого числа змінних та умов, досить важко знайти рішення дискретної транспортної задачі як рішення задачі лінійного програмування. Інколи доцільніше перевести її у вигляд задачі Моржа-Канторовича з дискретними ймовірнісними просторами, наприклад, на рисунку 1.5 . Тоді витратами на перетворення точок з одного розподілу в інший стануть відстані.

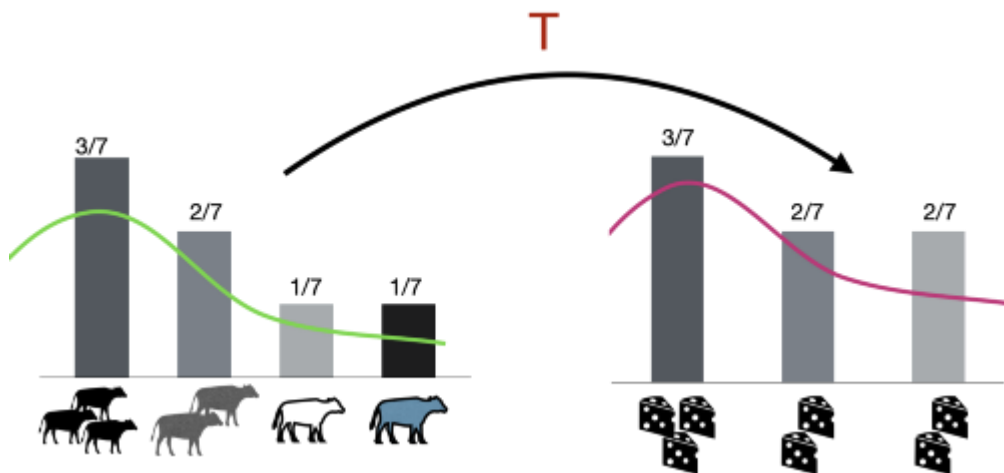


Рисунок 1.5 - Задача Монжа з дискретними розподілами ймовірностей

А вирішенням задачі буде мінімальна відстань між розподілами, що повноцінно переводить множину  $X$  в множину  $Y$ .

#### 1.4. Відстань Васерштайна

Метрика Канторовича, або відстань Васерштайна, це функція, що визначає відстань між ймовірнісними розподілами. Через аналогію з задачею про переміщення землі, ця метрика ще має назву відстань переміщення землі (англ. earth mover's distance).

Офіційно метрика була введена Р.Л. Дорбушиним в 1970 році, що посилався на роботи Л. Н. Васерштайна, що примінив її у вигляді:

$$W(\mu, \nu) = \inf_{\sigma \in \Pi(\mu, \nu)} \int_{X \times X} \varrho(x, y) \sigma(dx dy) \quad (1.7)$$

Проте, у 1939 році Леонід Канторович в роботі “Математичний метод планування і організації виробництва” [8] визначив схожу  $L^p$ -метрику:

$$W_p(\mu, \nu) = \inf_{\sigma \in \Pi(\mu, \nu)} \left( \int_{X \times X} \varrho(x, y)^p \sigma(dx dy) \right)^{1/p} \quad (1.8)$$

де,  $\Pi(\mu, \nu)$  - множина всіх мір на  $X \times X$  таких, що дві випадкові величини  $X_1$  та  $X_2$ , що визначені на ймовірнісних просторах з мірами  $\mu$  та  $\nu$ , то існує такий ймовірнісний простір з мірою  $\rho \in \Pi(\mu, \nu)$  на якому буде існувати дві випадкові величини  $Y_1$  та  $Y_2$  такі, що  $Y_1$  буде мати такий самий розподіл, як  $X_1$ , одночасно з  $Y_2$  матиме розподіл як  $X_2$ .

Очевидно, що  $W_p(\mu, \nu) = W_p(\nu, \mu)$  та  $W_p(\mu, \nu) = 0$  лише тоді, коли  $\mu = \nu$ .

Метрика Васерштайна може бути еквівалентно визначена через математичне сподівання:

$$W_p(\mu, \nu) = (\inf E(\varrho(X, Y)^p))^{1/p} \quad (1.9)$$

де  $E(Z)$  - це математичне сподівання випадкової величини  $Z$ , а інфімум береться на спільному розподілі випадкових величин  $X$  та  $Y$  з маргіналами (відособленнями)  $\mu$  та  $\nu$  відповідно.

Повертаючись до задачі оптимального транспортування, розподіл заводів виробництва  $\mu(x)$  на просторі  $X$  необхідно перевести у розподіл точок збуту  $\nu(y)$  на тому ж просторі. Для двох одновимірних розподілів відстань Васерштайна можна зобразити через сумісний транспортний план як на рисунку 1.6 на просторі  $xOy$ .

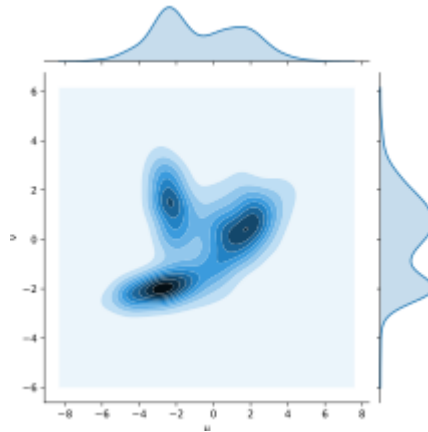


Рисунок 1.6 - Сумісний план транспортування для двох одновимірних розподілів

Функція вартості  $\varrho(x, y)$  напівнеперервна знизу, тобто достатньо лише неперервності поблизу мінімуму функції. Транспортний план перевезення товарів з точок виробництва в точки збуту можна описати функцією  $\sigma(x, y)$ , що задовольняє наступним умовам:

$$\begin{aligned} \int_X \sigma(x, y) dy &= \mu(x) \\ \int_X \sigma(x, y) dx &= \nu(y) \end{aligned} \quad (1.10)$$

Тобто підприємствам не дозволяється виробляти більше товарів, аніж потрібно для збуту тим самим роблячи задачу замкненою.

$$\int_X \int_X \varrho(x, y) \sigma(x, y) dx dy = \int_{X \times X} \varrho(x, y) d\sigma(x, y) \quad (1.11)$$

Транспортний план буде оптимальним при мінімальній функції вартості, тобто такий, що:

$$W(\mu, \nu) = \inf_{\sigma \in \Pi(\mu, \nu)} \int_{X \times X} \varrho(x, y) \sigma(dx dy) \quad (1.12)$$

При умові, що вартість переміщення це відстань між двома точками - оптимальний транспортний план відповідає метриці Канторовича при  $p = 1$  або метриці Васерштайна.

З точки зору чисельних методів, - це лінійна програма, яка завжди існує і може бути вирішена в  $O(n^3)$  час дискретних  $\mu, \nu$ .

Метрика Васерштайна часто використовується для порівняння дискретних розподілів, наприклад, в кольорових гістограмах двох цифрових зображень, порівнянням гістограм двох хвороб для визначення їх схожості, розподіл симптомів хвороби пацієнта та знаходження хвороби з найбільш схожим набором симптомів.

### 1.5. Незбалансовані задачі оптимального транспортування

Так як в реальному житті задачі транспортування дуже нечасто є збалансованими (рисунок 1.7), тому існує узагальнення на формулювання задачі Канторовича для послаблення граничних обмежень шляхом додавання дивергенту щоб врахувати зміну маси.

Розглянемо більш загальний випадок — стохастичний (тобто відображення один до багатьох), тому що це найбільш підходяща модель для багатьох практичних завдань. Наприклад, в клітинній біології природно думати, що одна клітина з вихідної популяції потенційно дає початок безлічі клітин в цільовій популяції. Введемо ймовірнісний простір  $(Z, \mathcal{B}(Z), \lambda)$  що є джерелом випадковості на транспортній карті  $T$ . На практиці можна взяти  $Z = \mathbb{R}^n$  і  $\lambda$  в якості стандартної гаусівської міри, якщо потрібно стохастическое відображення; в іншому випадку для  $\lambda$  можна встановити детермінований розподіл.

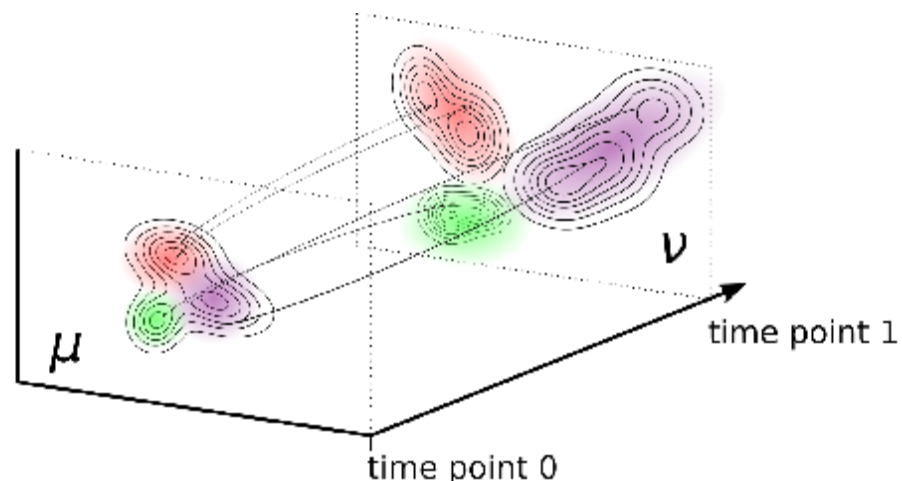


Рисунок 1.7 - Незбалансована задача оптимального транспортування

Введемо 2 функції зміни маси  $c_1: X \times Y \rightarrow R$  та  $c_2: R \rightarrow R$  та збалансування задачі. Для функції  $c_1$  використаємо відображення  $T$ , та для функції  $c_2$  введемо відображення  $\xi: X \rightarrow R$  як коефіцієнт масштабування, що фіксує зміну маси. Тоді для знаходження відображень необхідно мінімізувати:

$$L(\mu, \nu) = \inf_{T, \xi} \int_X \int_Z c_1(x, T(x, y)) d\lambda(z) \xi(x) d\mu(x) + \int_X c_2(\xi(x)) d\mu(x) \quad (1.13)$$

при обмеженні  $T(\xi\mu \times \lambda) = \nu$ .

Для того, щоб послабити рівність в обмеженні, додають дивергент (div):

$$L(\mu, \nu) = \inf_{T, \xi} \int_X \int_Z c_1(x, T(x, y)) d\lambda(z) \xi(x) d\mu(x) + \int_X c_2(\xi(x)) d\mu(x) + D_\psi(T(\xi\mu \times \lambda) | \nu), \quad (1.14)$$

де  $D_\psi$  -  $\psi$ -дивергент між невід'ємними скінченними мірами  $P, Q$  на  $T \subseteq R^d$  індукованою нижньою напівнеперервною, опуклою функцією ентропії  $\psi: R \rightarrow R \cup \{\infty\}$ , що визначається як:

$$D_\psi(P, Q) = \psi_\infty' P^\perp(T) + \int_T \psi\left(\frac{dP}{dQ}\right) dQ, \quad (1.15)$$

де  $\psi_\infty' = \lim_{s \rightarrow \infty} \frac{\psi(s)}{s}$  та  $\frac{dP}{dQ} Q + P^\perp(T)$  є розкладом Лебега міри  $P$  щодо  $Q$ .

Дивергент можна переписати з використанням протилежної функції  $f: Y \rightarrow (-\infty, \psi_\infty']$  у вигляді:

$$D_{\psi}(T(\xi\mu \times \lambda)|\nu) = \sup_f \int_X \int_Z f(T(x,z)) d\lambda(z) \xi(x) d\mu(x) - \int_Y \psi^*(f(y)) d\nu(y) \quad (1.16)$$

де  $\psi^*$  - опукла спряжена функція  $\psi$ .

Тепер цільова функція може бути оптимізована за допомогою чергування стохастичних градієнтів після параметризації  $T$ ,  $\xi$  та  $f$  з нейронними мережами.

З точки зору чисельних методів, сучасний спосіб вирішення збалансованих задач оптимального транспортування - це клас ітеративних алгоритмів масштабування, які узагальнюють алгоритм Сінкхорна для обчислення регуляризованих транспортних планів. Не існує практичних алгоритмів для незбалансованого оптимального транспортування між безперервними мірами, особливо у просторах високих розмірів.

Для знаходження рішення необхідно мінімізувати невикуклу цільову функцію і вивчити транспортний план  $T$  та коефіцієнт масштабування  $\xi$ . Проте через неопуклості цільової функції оптимізації алгоритм не гарантує знаходження глобального оптимуму.

Натомість немасштабована задача оптимального транспортування буде опуклою та гарантує знаходження глобального оптимуму:

$$W_{unb}(\mu, \nu) = \inf_{\gamma} \int_{X \times Y} c(x,y) d\gamma(x,y) + D_{\psi_1}(\pi_{\#}^X \gamma | \mu) + D_{\psi_2}(\pi_{\#}^Y \gamma | \nu), \quad (1.17)$$

де  $\pi^X$  та  $\pi^Y$  - оператори проектування на  $X$  та  $Y$  відповідно, а  $\pi_{\#}^X$  та  $\pi_{\#}^Y$  - маргінали спільної міри  $\gamma$ .

Проте це формулювання наразі є практичним лише для дискретних задач і має обмежену масштабованість.

## 1.6. Аналіз Прокруста

Метрика має певний зв'язок з аналізом Прокруста, що визначає, як треба перетворити розподіл, щоб він був ідентичним до другого розподілу. Певні етапи перетворення указані на рисунку 1.8.

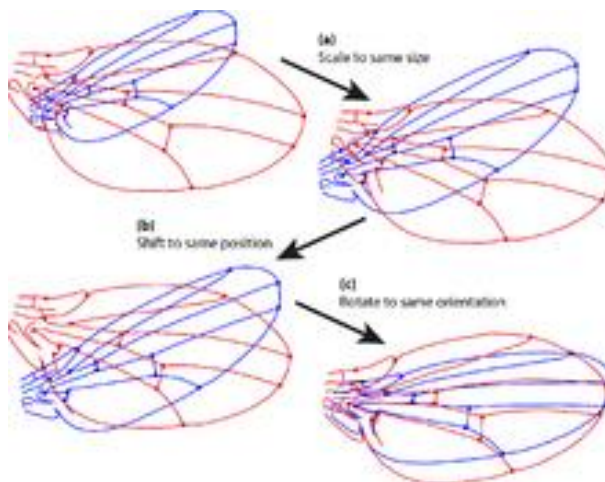


Рисунок 1.8 - Три етапи перетворення Прокруста

Аналіз Прокруста у вигляді мір хіральності та аналізу форми допомагає зрозуміти наскільки сильно і як саме необхідно змінити один або інший розподіл, щоб відстань між ними була меншою.

Хіральність означає здатність розподілу мати своє дзеркальне відбиття, що не дорівнює оригіналу. Два розподіли можуть мати велику відстань Канторовича проте бути хіральною, тобто віддзеркаленими. При знаходженні хірального центру або осі хіральності можливо знайти таке перетворення розподілу, що дасть найменшу відстань.

Щоб порівняти форми двох або більше об'єктів, об'єкти повинні бути оптимально «розміщеними». У аналізі Прокруста оптимальним накладання об'єкта виконується переносом, обертанням та зміною масштабу. Ціль накладання постає в тому, щоб зменшити відстань Прокруста між об'єктами.

Тож задача знаходження схожості форм розподілів буде полягати у знаходженні мінімуму функції:

$$\min_{P_x, P_y} ||P_x X - k P_y Y||^2 \quad (1.18)$$

де,  $P_x$  та  $P_y$  - матриці перетворення для розподілу  $X$  та  $Y$  відповідно.

Узагальнений аналіз Прокруста це статистичний метод, що можна використовувати для порівняння форм об'єктів, гістограм розподілу, результатів опитувань, інтерв'ю при умові невідповідності двох розподілів, форм, відповідей одному виду.

Алгоритм аналізу має вигляд поетапного порівняння відстаней Прокруста до певної еталонної опорної форми. Тим самим, повертаючись до транспортної задачі зі бізнес-аналізу можна визначити найкращий спосіб розміщення тих чи інших точок в залежності від вже розміщених точок збуту або точок виробництва. Аналіз Прокруста використовують як комбінацію з іншими методами, наприклад знаходженням відстані Васерштайна [9]

### 1.7. Висновки до розділу

В першому розділі було формулювання задачі оптимального транспортування у вигляді задачі лінійного програмування, а також у вигляді задачі знаходження відображення одного розподілу в інший. Детально було розглянуто сфери використання транспортної задачі, актуальність та різноманітність. В той час як маломасштабні та збалансовані задачі можуть бути вирішені чисельними або графічними методами, то зі зростанням складності, а відповідно зі зростанням

відповідності задачі до реального життя, ці методи вже не є використовуваними. Тому постає актуальне питання про вирішення задачі оптимального транспортування іншими методами. Існуючі методи вирішення будуть розглянуті у наступному розділі.

## РОЗДІЛ 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ

### 2.1. Вступ

В другому розділі буде розглянуто існуючі методи вирішення задачі оптимального програмування як у вигляді задачі лінійного програмування, так і у вигляді задачі мінімізації відстані між розподілами.

Будуть розглядатися саме методи вирішення дискретних задач як найбільш часто використовуваних та наближених до задач з реального життя.

Вирішення задачі у вигляді задачі лінійного програмування корисне та зручне при достатньо малій кількості точок в обох множинах, коли задачу можна представити у вигляді матричної таблиці.

Розглядаються як відкриті так і закриті транспортні задачі.

### 2.2. Лінійне програмування

Задачу оптимального транспортування можна представити у вигляді задачі лінійного програмування у вигляді:

$$\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \rightarrow \min$$

При умовах:

$$\begin{aligned} \sum_{j=1}^m x_{ij} &\leq a_i, a_i \in X \\ \sum_{i=1}^n x_{ij} &= b_j, b_j \in Y \end{aligned} \tag{2.1}$$

$$x_{ij} \geq 0$$

Збалансованість задачі є необхідною і достатньо умовою існування допустимих розв'язків задачі. При незбалансованості задачі за умови незадоволення попиту пропозицією вводиться множина штрафів за недоставку товарів в певний пункт і задача зводиться до мінімізації суми витрат як штрафних, так і транспортувальних. Таку задачу можна звести до збалансованої додаванням додаткової фіктивної точки виробництва з транспортними витратами що дорівнюють штрафним. У іншому випадку при перевиконанні попиту задачу зводить до збалансованої, або закритої, додаванням ще однієї фіктивної точки збуту з об'ємом попиту, що дорівнює різниці між реальним попитом та реальною пропозицією.

Збалансована транспортна задача, як на рисунку 2.1, завжди має оптимальний розв'язок через обмеженість і замкнутість, тобто компактність, її області визначення, а функція, що мінімізується, є неперервною і досягає свого глобального мінімуму на ньому.

Існують методи знаходження опорного плану як базисного допустимого розв'язку задачі лінійного програмування [11]. Метод північно-західного кута базується на заповненні лівого верхнього кута транспортної таблиці і в залежності від значень запасу і потреби викреслюють або стовпець або рядок транспортної таблиці і покладаючи матрицю розв'язку як мінімум між запасами та потребами в цій клітині. Початковий план перевезень знаходиться не більше ніж за кроків, на кожному з яких у транспортній таблиці заповнюють одну комірку, яку, в подальшому, називають зайнятою.

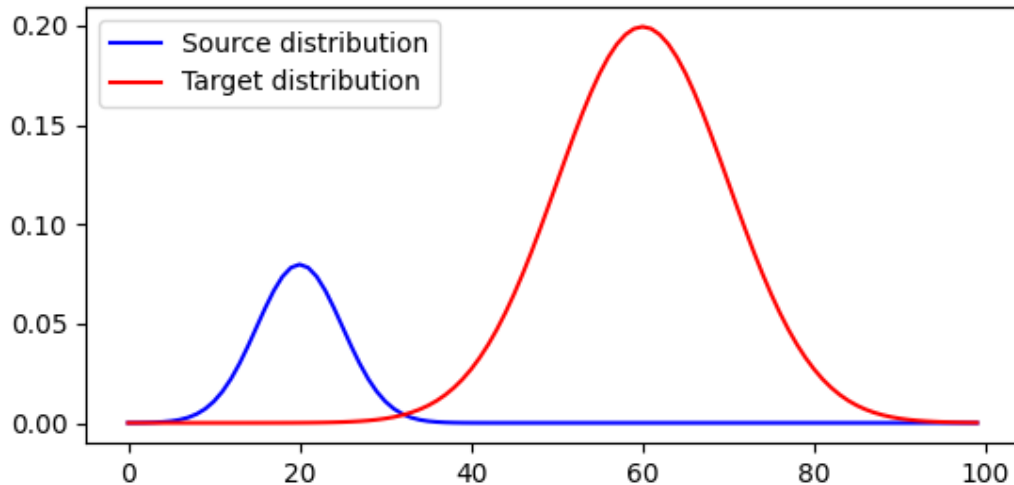


Рисунок 2.1 - Незбалансована транспортна задача

Недоліком цього методу є повне ігнорування матриці транспортних витрат на етапі пошуку опорного плану.

Метод мінімального елемента відрізняється від методу північно-західного кута лише на етапі вибору послідовності заповнення клітинок таблиці. В цьому методі за активну клітинку вибирається та, в якій витрати транспортування є мінімальними.

Рішення симплекс-методом, (рисунок 2.2) або методом поступового покращення плану, задачі лінійного програмування полягає у керованому русі по опорних планах до оптимального. Він дозволяє знайти оптимальне рішення задачі при наявності початкового опорного плану.

Метод потенціалів є модифікуванням симплекс-методу і використовує другу теорему двоїстості для перевірки оптимальності. Для кожної комірки транспортної таблиці ставляться у відповідність сума двох потенціалів. Одному з усіх потенціалів задається довільне значення, а інші визначаються відповідно до нього. Якщо всі ці числа є від'ємними, або нулями, то опорний план є оптимальним, в протиріччя знаходиться найбільше додатне значення і його змінна вводиться в

базис, компоненти нового опорного плану розраховуються за формулою симплекс-методу.

| Базис    | $x_0$    | $x_1$    | $x_2$    | ...      | $x_k$    | ...      | $x_m$    | $x_{m+1}$      | ...      | $x_p$      | ...      | $x_n$      |
|----------|----------|----------|----------|----------|----------|----------|----------|----------------|----------|------------|----------|------------|
| $x_1$    | $b_1$    | 1        | 0        | ...      | 0        | ...      | 0        | $a_{1,m+1}$    | ...      | $a_{1,p}$  | ...      | $a_{1,n}$  |
| $x_2$    | $b_2$    | 0        | 1        | ...      | 0        | ...      | 0        | $a_{2,m+1}$    | ...      | $a_{2,p}$  | ...      | $a_{2,n}$  |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$       | $\vdots$ | $\vdots$   | $\vdots$ | $\vdots$   |
| $x_k$    | $b_k$    | 0        | 0        | ...      | 1        | ...      | 0        | $a_{k,m+1}$    | ...      | $a_{k,p}$  | ...      | $a_{k,n}$  |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$       | $\vdots$ | $\vdots$   | $\vdots$ | $\vdots$   |
| $x_m$    | $b_m$    | 0        | 0        | ...      | 0        | ...      | 1        | $a_{m,m+1}$    | ...      | $a_{m,p}$  | ...      | $a_{m,n}$  |
|          | $F$      | 0        | 0        | ...      | 0        | ...      | 0        | $\Delta_{m+1}$ | ...      | $\Delta_p$ | ...      | $\Delta_n$ |

Рисунок 2.2 - Симплекс-таблиця

Зациклення методу потенціалів хоч і мало ймовірно та все ж можливе, тому необхідно застосовувати правило, що гарантуватиме однозначність виборів базису при однакових максимальних значеннях.

Основна проблема цього методу це перевірка оптимальності. Задля пришвидшення цього процесу використовують бар'єри, циклічний перегляд або використання декількох претендентів.

Цим методом при невеликих модифікаціях можна скористатися при додаванні обмеження на пропускну здатність або при багатопродуктовості задачі.

Вирішення задачі оптимального транспортування через знаходження оптимального плану транспортної таблиці є зручним і наглядним методом проте лише при невеликій кількості точок. При їх зростанні збільшиться складність та зменшиться швидкість обробки даних, навіть при чисельному методі обрахунку використовуючи комп'ютерну обробку. При збільшенні розмірності множин, вирішення задачі такими методами стає майже неможливим.

### 2.3. АНА метод

За аббревіатурою АНА стоїть шифрування фамілій Ауренхаммера, Гофмана та Аронова (англ. F. Aurenhammer, F. Hoffmann and B. Aronov), що у 1998 розглянули алгоритм для пошуку оптимального транспортного плану у статті “*Minkowski-Type Theorems and Least-Squares Clustering*” [12]. Ідея алгоритму постає у розгляданні транспортної задачі з евклідовою квадратичною функцією вартості як еквівалент необмеженій та неперервній задачі мінімізації для певної опуклої цільової функції. Цей алгоритм також реалізований для випадку багатомасштабної задачі та однаково працює в одномасштабному варіанті.

Використання методу найменших квадратів може бути корисним при вирішенні певної транспортної задачі і при знаходженні апроксимації методом найменших квадратів двох наборів точок, де дозволені переміщення і масштабування.

Цей метод застосовується до напівдискретного випадку в  $R^d$  з квадратичною евклідовою відстанню:

$$d(x, y) = ||x - y||^2 \quad (2.2)$$

Також вважається, що абсолютно неперервна по мірі Лебега, а  $\mathcal{U}$  — кінцева сума Дірака, або сума зважених дельта-функцій Кронекера:

$$\mathcal{U} = \sum_{j=1}^n \mathcal{U}_j \delta_{y_j} \quad (2.3)$$

Ключова ідея методу АНА полягає у представлення будь-якої діаграми потужності, або мозаїки Лагерра (англ. Laguerre tessellation) (рисунок 2.3), що регулюється точками опори  $y_1, y_2, \dots, y_n$  з  $\mathcal{U}$  та довільно визначенні ваги  $w_1, w_2, \dots, w_n \in R$  характеризує оптимальний транспортний план від міри  $\mu$  до певної міри, о знаходиться на цих точках опори.

(b) Laguerre tessellation

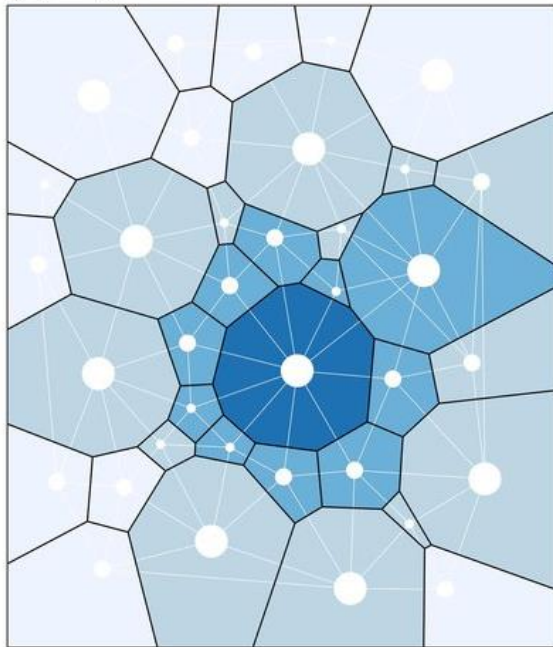


Рисунок 2.3 - Мозаїка Лаггера

Мінімізуючи функцію за рахунок зміни вагів ми можемо знайти діаграму потужності, що визначає оптимальне транспортування до коректної міри  $\mathcal{U}$ .

Діаграма потужності визначається як відповідність точки  $y_j$  до ваги  $w_j$ . Кожна комірка, що належить до  $y_j$  відносно вектора з вагів  $w_n \in R^n$  визначається множиною:

$$Pow_j(w) = \{x \in R^d: ||x - y_j||^2 - w_j \leq ||x - y_i||^2 - w_i, i = \overrightarrow{1, n}\} \quad (2.4)$$

Перетин двох сусідніх клітин є гіперплощиною, де вищезазначена нерівність перетворюється у рівність і таким чином визначається для  $\mu$ нульова множина. Геометрично гіперплощини можна охарактеризувати так: якщо провести сфери навколо точок  $y_j$ , радіуси яких є квадратичними коренями ваг  $w_j$ , то гіперплощина містить перетин сфер, при умові, що він не порожній. Якщо сфери торкаються, то гіперплощина є загальною дотичною площиною в точці перетину. Якщо сфери не перетинаються, то гіперплощина визначається як множина точок в  $R^d$ , відстань яких до точок дотику двох сфер є рівні, як на рисунку 2.4.

Транспортна карта  $T_w$ , що відображає елемент  $x \in R^d$  в точку  $y_j$  так що  $x \in Row_j(w)$  визначено  $\mu$  майже всюди. Отже задача полягає у знаходженні такого вектору  $w$ , що  $T_w$  буде оптимальною. Знайти такий вектор можна при мінімізації функції:

$$F(w) = \sum_{j=1}^n [\mathcal{U}_j * w_j - \int_{Row_j(w)} (||x - y_j||^2 - w_j d\mu(x)) ] \quad (2.5)$$

Обчислення функції  $F(w)$  по заданому ваговому вектору  $w_n \in R^n$  полягає обрахування діаграми потужності та доволі просте інтегрування по кожній комірці. Функція  $F(w)$  диференційовна і його  $i$ -тий компонент це фактична різниця між  $\mathcal{U}_i$   $\mu$ -масою, що перенесли в цю точку згідно діаграмі потужності. Кожна діаграма потужності визначає карту Монжа  $\mu$ -майже скрізь як кусково-постійну карту, яка відображає внутрішність кожної комірки  $Row_j(w)$  на  $y_j$ .

Для оптимізації функції  $F(w)$  використовують квазіньютонівський метод з обрахуванням оберненого гесіану (визначник матриці Гессе) або метод градієнтного спуску. Нещодавно, у 2017 році, для оптимізації цієї функції був використаний затухаючий алгоритм Ньютона, що показав непогані результати [13].

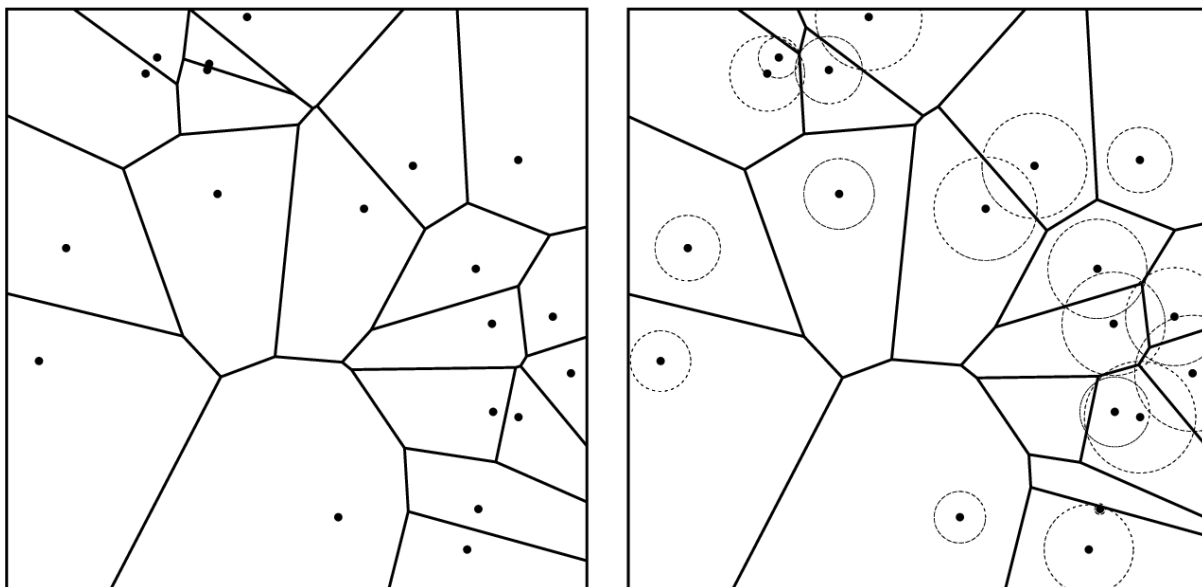


Рисунок 2.4 - Діаграми центральної потужності

Метод АНА можна використовувати у випадку неквадратичної евклідової відстані ( $p=1$ ), проте з'являється проблема, що границі клітинок більше не є гіперплощинами, що робить обчислення функції та інтегралу більш витратним. Метод є достатньо дієвим, хоча і накладає обмеження на простір та міри.

#### 2.4. Регульоване оптимальне транспортування. Алгоритм масштабування Сінкхорна та Грінхорна

Ідея про регулювання задачі оптимального транспортування через ентропійний штраф з'явилися ще у 1969 році, проте у 2013 році М. Куруті у своїй роботі виявив, що ентропійно регульовані задачі оптимального транспортування можуть бути дуже ефективно вирішені за допомогою простого алгоритму матричного масштабування [14]. Цей алгоритм іноді називають масштабуванням Сінкхорна або

алгоритмом Сінкхорна-Кноппа (англ. or Sinkhorn-Knopp Algorithm ). Це одна з ключових причин, чому ентропія використовується як термін регуляризації замість інших варіантів.

Для пари елементів  $\sigma$  член ентропійної регуляризації буде мати вигляд:

$$H(\sigma) = - \sum_{i,j} \sigma_{i,j} (\log(\sigma_{i,j}) - 1) \quad (2.6)$$

При введенні параметра регуляризації  $\lambda > 0$ , ентропійно регуляризована задача оптимального транспортування буде мати вигляд:

$$\min_{\sigma \in \Pi(\mu, \nu)} \sum_{i,j} c_{i,j} \sigma_{i,j} - \lambda H(\sigma) \quad (2.7)$$

Так як ентропія - строго увігнута функція, то регуляризована задача оптимального транспортування буде строго опуклою. Отже, існує єдине рішення  $\sigma^*$  для кожного  $\lambda > 0$ . Рішення такої задачі завжди буде щільним, на відміну від оригінальної задачі, що вирішується за допомогою лінійного програмування, де вектор рішень зазвичай є розрідженим (вектор, який має велику кількість нулів, тому він займає небажаний простір для зберігання цих нулів).

При малих  $\lambda$  задача подібна до класичної, при великих — задача буде більш опуклою.

$$\lim_{\lambda \rightarrow 0} \sigma^*_\lambda = \sigma^*, \quad (2.8)$$

де  $\sigma^*$  - оптимальне рішення оригінальної задачі транспортування.

$$\lim_{\lambda \rightarrow \infty} \sigma^*_{\lambda} = \mu \otimes \nu \quad (2.9)$$

Відстань Васерштайна з регуляризованою ентропією буде мати вигляд наступної формули, що можна розглядати як наближення до оригінальної відстані Васерштайна при малих  $\lambda$ :

$$W_{p,\lambda}(\mu, \nu) = (\inf_{\sigma \in \Pi(\mu, \nu)} (\int_{X \times X} \varrho(x, y)^p \sigma(dx dy) ) - \lambda H(\sigma))^{1/p} \quad (2.10)$$

Для фіксованого значення  $\lambda > 0$  ядро Гіббса (Вибірка Гіббса породжує марківський ланцюжок зразків, кожен з яких корелює з найближчими зразками), що асоціюється з матрицею витрат  $C$  це таке  $K \in R^{n \times m}$ , що:

$$K_{i,j} = e^{\frac{c_{ij}}{\lambda}}, i = \overrightarrow{1, n}, j = \overrightarrow{1, m} \quad (2.11)$$

Основна ідея алгоритму масштабування Сінкхорна полягає у представленні єдиного рішення  $\sigma^*_{\lambda}$  у вигляді  $\sigma^*_{\lambda, i, j} = u_i K_{i, j} v_j$  з векторами  $u \in R^n, v \in R^m$  такі що задовольняють обмеженням:

$$\begin{aligned} \text{diag}(u)Kv &= \mu \\ \text{diag}(v)K^T u &= \nu, \end{aligned} \quad (2.12)$$

де  $\text{diag}(u)$  - матриця  $n \times n$  з вектором  $u$  на діагоналі та нулях на інших місцях.

Загальна ідея для знаходження векторів  $u$  та  $v$ , що задовольняють обмеженням - це використання ітеративного методу з початковим довільним позитивним вектором, наприклад вектор з одиниць, та по

черзі оновлювати вектори задля досягнення умов. Це досягається ітераціями:

$$\begin{aligned} u^{(k+1)} &= \frac{\mu}{Kv^{(k)}} \\ v^{(k+1)} &= \frac{\nu}{K^t u^{(k)}} \end{aligned} \quad (2.13)$$

По своїй суті це метод альтернативних проекцій: : поточне рішення поперемінно проектується на множини рішень  $(u, v)$ , що задовольняють обмеженням. Ітерації транспортних матриць при  $k \rightarrow \infty$  прямує до оптимального рішення( формула 2.14). Алгоритм у формі псевдокоду указаний на рисунку 2.5.

$$\sigma^{(k)}_{\lambda} = \text{diag}(u^{(k)})K\text{diag}(v^{(k)}) \rightarrow_{k \rightarrow \infty} \sigma^*_{\lambda} \quad (2.14)$$

---

**Algorithm 1** Sinkhorn-Knopp Algorithm (SK).

---

**Require:**  $\mathbf{a}, \mathbf{b}, \mathbf{C}, \lambda$

$\mathbf{u}^{(0)} = \mathbf{1}, \mathbf{K} = \exp(-\mathbf{C}/\lambda)$

**for**  $i$  in  $1, \dots, n_{it}$  **do**

$\mathbf{v}^{(i)} = \mathbf{b} \oslash \mathbf{K}^T \mathbf{u}^{(i-1)}$  // Update right scaling

$\mathbf{u}^{(i)} = \mathbf{a} \oslash \mathbf{K} \mathbf{v}^{(i)}$  // Update left scaling

**end for**

**return**  $\mathbf{T} = \text{diag}(\mathbf{u}^{(n_{it})})\mathbf{K} \text{diag}(\mathbf{v}^{(n_{it})})$

---

Рисунок 2.5 - Алгоритм Сінкхорна в псевдокодi

На практиці алгоритм має хороші показники виконання по часу збіжності, його використовують в сукупності з іншими методами, де необхідні масштабні обчислення або наближення оптимальних транспортних відстаней, наприклад, при обчисленні барицентрів Васерштайна для інтерполяції форми. У 2013 році М. Кутурі вже дав

експериментальні докази, що використання Сінкхорна для вирішення задачі оптимального транспортування перевершує найсучасніші на той час методи.

Проте алгоритм має свої мінуси, зокрема числової стабільності. Наближення відстані Васерштайна краще при малому значенні параметра регуляризації. Проте у виборі малого значення  $\lambda$  є певні недоліки:

- Мале значення параметру регуляризації  $\lambda$  призводить до проблеми зі слабкою опуклістю. Отже, збіжність  $\sigma^{(k)}_{\lambda} \rightarrow_{k \rightarrow \infty} \sigma^*_{\lambda}$  повільніше.
- Елементи ядра Гіббса швидко наближаються до нуля при малому  $\lambda$ . Це призводить до чисельно нестабільних ітерацій масштабування матриці, що є особливо проблематичним, коли  $K$  наближається або опускається нижче порога точності машини. Це означає, що  $\lambda$  не можна вибрати довільно малим.

Однак існують способи вирішення цих проблем, такі як  $\lambda$ -масштабування або стабілізація в лог-області [15]. Ще один нещодавно запропонований (у 2017 році) варіант масштабування Сінкхорна оновлює лише один жадібно вибраний рядок або стовпець на кожному кроці і тому називається Грінхорн (англ. Greenkhorn) [16].

Замість того, щоб виконувати почергове оновлення всіх рядків і стовпців  $K$ , алгоритм Грінхорна оновлює лише один рядок або стовпець на кожному кроці. У статті “*Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration*”[22] наведені експериментальні докази про перевагу алгоритму Грінхорна над алгоритмом Сінкхорна (рисунок 2.6).

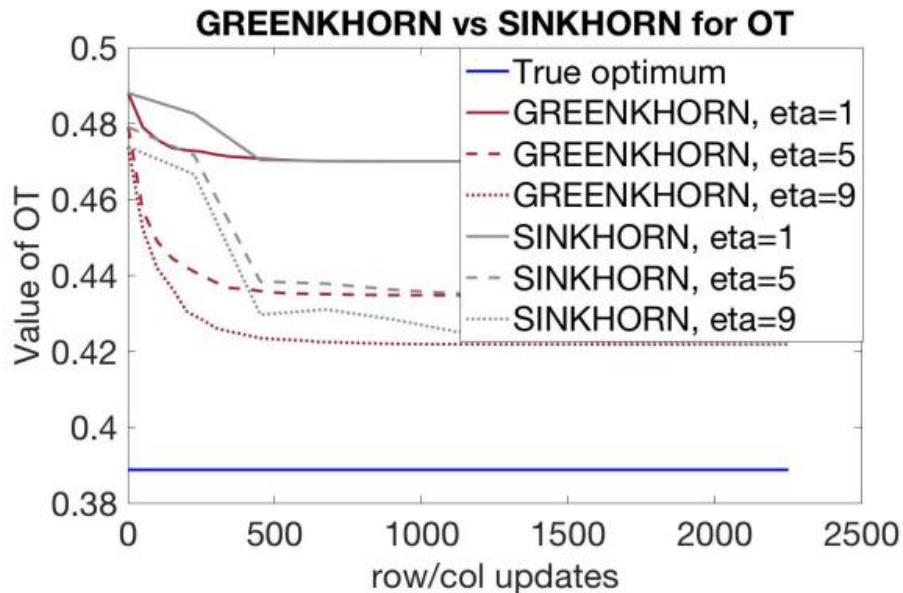


Рисунок 2.6 - Порівняння алгоритмів Сінкхорна та Грінхорна

Багатомасштабні підходи до вирішення задач оптимального транспортування з'являються в статтях та роботах для напівдискретної транспортної проблеми щонайменше з 2010 р.

Для задачі багатомасштабного оптимального транспортування розглянемо міри  $\mu$  та  $\nu$  та кінцеві опори  $X = \text{supp}(\mu)$ ,  $Y = \text{supp}(\nu)$  та функція витрат  $c: X \times Y \rightarrow \mathbb{R}_+$  записаної у вигляді матриці витрат. Багатомасштабний метод можна розділити на три кроки (рис.11):

- Огрубіння (Coarsening): З початкового екземпляра створюється кінцева послідовність екземплярів з меншою масштабністю. Це вимагає побудови ієрархічних розділів для множин  $X$  і  $Y$  та визначення функції витрат (матриці) на кожній шкалі.
- Поширення (Propagation): Якщо ми маємо здійснений транспортний план  $\pi(i)$  у певному масштабі і методи розповсюдження будують можливий план  $\pi(i-1)$  на більш тонкому масштабі  $i-1$ .

- Доопрацювання (Refinement): Транспортний план ітеративно вдосконалюється до оптимальності. Стратегії уточнення зазвичай діють в одному масштабі.

Алгоритм розв'язку рішення для багатомасштабних задач оптимального транспортування указаний на рисунку 2.7.

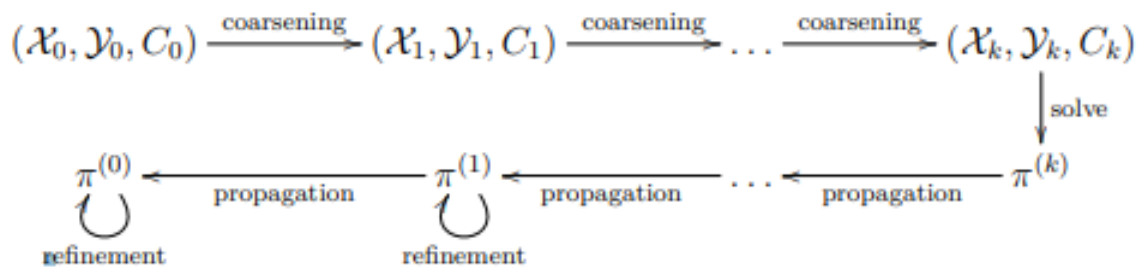


Рисунок 2.7 - Алгоритм розв'язку багатомасштабних задач

## 2.5. Машинне навчання для оптимізації функцій

Машинне навчання (англ. Machine learning) - це навчання комп'ютерних алгоритмів, які можуть автоматично вдосконалюватися за допомогою досвіду та за допомогою пошуку зв'язку між даними.

Підмножина машинного навчання тісно пов'язана з обчислювальною статистикою, яка зосереджена на прогнозуванні або знаходженні зв'язків між даними за допомогою комп'ютерів. Навчання математичної оптимізації поширює методи, теорію та сфери застосування у галузі машинного навчання.

Генеративні змагальні мережі, або скорочено GAN (англ. Generative Adversarial Network), являють собою підхід до генеративного моделювання з використанням методів глибокого навчання, таких як згорткові нейронні мережі.

Генеративное моделювання - це завдання навчання без учителя в машинному навчанні, яка включає в себе автоматичне виявлення і

вивчення закономірностей у вхідних даних таким чином, щоб модель можна було використовувати для створення або виведення нових прикладів, які, ймовірно, могли бути взяті з вихідного набору даних.

Дискримінаційна модель повинна розрізняти приклади вхідних змінних за класами; вона повинна вміти вибрати або прийняти рішення щодо того, до якого класу належить даний приклад. Неконтрольовані моделі, які підсумовують розподіл вхідних змінних, можуть бути використані для створення або генерації нових прикладів у вхідному розподілі. Комбінування таких моделей і є генеративна змагальна мережа.

Тобто структуру мережі можна розкласти на дві частини (рисунок 2.8):

- Генератор. Модель, яка використовується для створення нових правдоподібних прикладів з проблемної області.
- Дискримінатор. Модель, яка використовується для класифікації прикладів як справжніх (з предметної області) або підроблених (згенерованих).

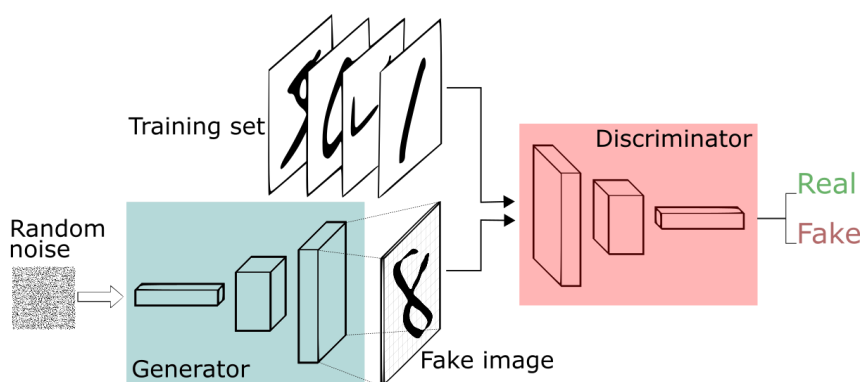


Рисунок 2.8 - Модель генеративної змагальної мережі.

Модель генератора приймає випадковий вектор фіксованої довжини в якості вхідних даних і генерує вибірку в області. Дискримінатор - це звичайна модель класифікації. Після процесу

навчання модель дискримінатора відкидається, оскільки нас цікавить генератор.

Розуміючи роль генератора та дискримінатора, мережу можна використовувати для генерації таких наборів з одного простору, що вони будуть максимально схожі на інший. Тоді відображення, що буде переводити один набір в інший і буде вирішенням задачі транспортування.

Для вирішення задачі оптимального транспортування генеративну змагальну мережу можна інтерпретувати як змагальну гру між відображенням  $T: X \rightarrow Y$  та  $f$ . Для покращення результатів використовується алгоритм масштабування Сінкхорна та Грінхорна.

Алгоритм для оптимізації цільової функції представлений у вигляді:

- $T$  бере точку  $x \sim \mu$  і транспортує її з  $X$  у  $Y$  шляхом породження  $T(x, z)$  де  $z \sim \lambda$ .
- $\xi$  визначає важливість ваги кожної транспортованої точки.
- Їх спільною метою є мінімізація розбіжностей між транспортованими зразками та реальними зразками від  $v$ , що вимірюється противником  $f$ .
- Крім того, функції витрат  $c_1$  і  $c_2$  спонукають  $T$ ,  $\xi$  до пошуку найбільш економічно ефективної стратегії.

## 2.6. Використання дерев для незбалансованих задач

Для незбалансованих задач було запропоновано декілька методів, серед них узагальнена відстань Канторовича-Рубенштейна (англ. *Generalized Kantorovich Rubinstein Distance*, *GKR*), що виражається у вигляді наступної формули:

$$GRK(\mu, \nu) = \min_{\gamma \in M(\mu, \nu)} \int_{X \times Y} c(x, y) d\gamma(x, y) + \quad (2.15)$$

$$+ \int_X \lambda_d d(\mu - \pi_{\#}^X \gamma) + \int_Y \lambda_c d(\nu - \pi_{\#}^Y \gamma),$$

де  $\lambda_d$  та  $\lambda_c: X \rightarrow R_+$  функції вартості для “руйнування” або “створення” відповідно.

Відстань GKR не обов’язково транспортує всю масу, але сплачує штрафи за створення та знищення маси. Важливо, що узагальнена відстань Канторовича-Рубенштейна включає багато популярних варіантів дистанції оптимального транспортування. Зауважимо, що функція витрат функція витрат є потужністю метрики  $L_p$   $c(x, y) = ||x - y||_p^p$ ), тоді відстань можна знайти за  $O(n^{2-\varepsilon})$  часу.

Існує теорема, про оцінювання відстані в 4-дереві:

$GKR_{euc}$  - відстань GKR з евклідовою вартістю,  $GKR_{tree}$  - це відстань GKR з квадратичною вартістю  $c(x, y) = d_T(x, y)$ , де  $T$  - чотиридереву (*quadtrees*). Існує така константа  $C$ , що:

$$GKR_{euc}(\mu, \nu) \leq C \cdot GKR_{tree}(\mu, \nu) \quad (2.16)$$

При тому, коли ми переводимо міру при конструюванні чотиридереву, то існує така оцінка для константи  $D$ :

$$E_T[GKR_{tree}(\mu, \nu)] \leq D \cdot GKR_{euc}(\mu, \nu) \log \Delta, \quad (2.17)$$

де  $\Delta$  - відношення найдалшої відстані до найближчої відстані.

Ця теорема вказує на те, що відстань GKR за евклідовою метрикою може бути наближена до відстані GKR за метрикою квадродерева. Формально проблему відстані ГКР на метриці дерева можна формалізувати таким чином:

Існує дерево  $T = (X, E, w)$  з  $n = |X|$  кількістю вузлів, функціями “руйнування” та “створення” та міри  $\mu, \nu$ .

Задачу знаходження відстані по дереву можна використовуючи алгоритм Сінкхорна або Грінхорна з лямбда, що визначається через машинне навчання. Проте в 2021 році був представлений новий, швидший спосіб обчислення узагальненої відстані Канторовича-Рубенштейна через деверо у роботі “Fast Unbalanced Optimal Transport on a Tree” , що здатен визначити відстань набагато швидше [20] (рисунок 2.9).

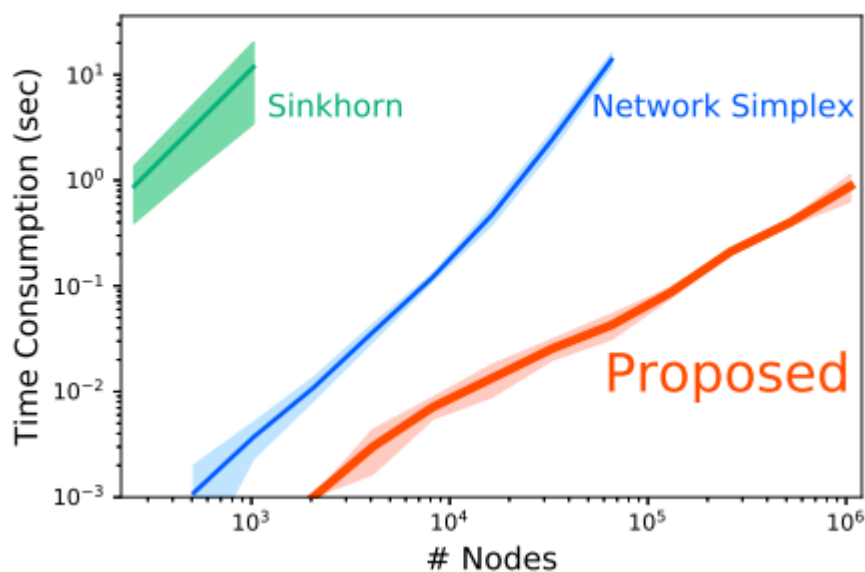


Рисунок 2.9 - Порівняння алгоритмів знаходження відстані

В основі цього алгоритму лежить динамічне програмування та прискорення обчислень за допомогою швидкої опуклої згортки з мінімізацією суми, ефективних структур даних та евристики зі зваженим об’єднанням. Динамічне програмування рекурсивно обчислює узагальнену відстань Канторовича-Рубенштейна від листка до кореня.

## 2.7. Висновки до розділу

У другому розділі було розглянуто існуючі методи вирішення транспортної задачі, в залежності від збалансованості, масштабності та просторів визначення.

Для складних незбалансованих та багатомасштабних задач методи, що шукають відстань або початковий та оптимальний опорний план стають дуже складними для розуміння, уяви та вирішення. Тому найкращим способом вирішення таких задач є побудова нейронної мережі, що здатна знаходити складні зв'язки між даними та будувати таке відображення, що буде щонайменше локально оптимальним.

## РОЗДІЛ 3. ПОБУДОВА ТА НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ. АНАЛІЗ РЕЗУЛЬТАТІВ

В даному розділі буде розглянута побудова нейронної мережі на основі генеративної змагальної мережі та перегляд результатів її роботи на кількох різноманітних прикладах.

### 3.1. Розгляд генеративної змагальної мережі та її адаптація.

#### Архітектура мережі

Генеративна змагальна нейронна мережа складається з двох різних нейронних мереж:

- Генеративна мережа — мережа, що створює елементи до навчального масиву по відмітним ознакам. На вхід в класичному варіанті подається вектор білого шуму.
- Дискримінаційна модель — мережа, що класифікує вхідний елемент до двох класів.

Така архітектура мережі дозволяє навчати її на нерозмічених даних, що дає змогу брати будь-який набір навчальних даних без їх попередньої особливої підготовки.

На вхід до генеративної мережі подається вектор білого шуму, що перетворюється по мірі навчання на один з елементів масиву. Проте в ситуації з задачею оптимального транспортування на вхід необхідно брати перший масив даних (у випадку з трансферним навчанням — нерозмічений) та перевіряти точність його перетворення на елементи другого масиву. Для забезпечення оптимальності розв'язку

використовується ентропійна регуляризація за допомогою алгоритма Грінхорна та відповідна функція витрат.

Для даної задачі визначимо архітектури генератора та дискримінатора (рис 3.1). на рисунках 3.1 . Вхідна форма даних для дискримінатора буде змінюватись в залежності від типу задачі, будь то зображення, координати чи одномірні розподілення. В той час, генератор може змінюватись повністю в залежності від типу та складності задачі.

Model: "Discriminator"

| Layer (type)               | Output Shape        | Param # |
|----------------------------|---------------------|---------|
| conv2d_3 (Conv2D)          | (None, 32, 32, 64)  | 1792    |
| leaky_re_lu_12 (LeakyReLU) | (None, 32, 32, 64)  | 0       |
| conv2d_4 (Conv2D)          | (None, 16, 16, 128) | 73856   |
| leaky_re_lu_13 (LeakyReLU) | (None, 16, 16, 128) | 0       |
| conv2d_5 (Conv2D)          | (None, 8, 8, 128)   | 147584  |
| leaky_re_lu_14 (LeakyReLU) | (None, 8, 8, 128)   | 0       |
| conv2d_6 (Conv2D)          | (None, 4, 4, 256)   | 295168  |
| leaky_re_lu_15 (LeakyReLU) | (None, 4, 4, 256)   | 0       |
| flatten (Flatten)          | (None, 4096)        | 0       |
| dropout (Dropout)          | (None, 4096)        | 0       |
| dense_3 (Dense)            | (None, 1)           | 4097    |
| Total params: 522,497      |                     |         |
| Trainable params: 522,497  |                     |         |
| Non-trainable params: 0    |                     |         |

Рисунок 3.1 - Архітектура змагальної мережі

Генеративна змагальна мережа має архітектуру, зображену на рисунку 3.2, що зображує об'єднання двох вже визначених мереж.

Model: "GAN"

| Layer (type)                  | Output Shape      | Param # |
|-------------------------------|-------------------|---------|
| Generator (Sequential)        | (None, 32, 32, 3) | 1007363 |
| Discriminator (Sequential)    | (None, 1)         | 522497  |
| Total params: 1,529,860       |                   |         |
| Trainable params: 1,007,363   |                   |         |
| Non-trainable params: 522,497 |                   |         |

Рисунок 3.2 - Архітектура генеративної змагальної мережі

В якості функції витрат для дискримінатора використовується бінарна крос-ентропія. Для генератора функція витрати визначається формулою 1.14 із застосування ентропійної регуляризації. визначеної формулою 2.7.

Метод оптимізації цільової функції є достатньо спільним для кожного типу задач, що і буде показано у розділі «Аналіз результатів».

### 3.2. Опис програмного забезпечення

Побудова та навчання нейронної мережі проводилось за допомогою мови програмування Python 3 та середовища Jupyter Notebook. Для візуалізації результатів було використане візуальне доповнення до бібліотеки tensorflow — TensorBord.

При побудуванні були використанні відкриті бібліотеки приведені в таблиці 3.1.

Таблиця 3.1 — Опис необхідних бібліотек Python 3

| Назва бібліотеки | Версія | Короткий опис                                        |
|------------------|--------|------------------------------------------------------|
| pumpy            | 1.21.4 | Попередня обробка та маніпуляції з даними            |
| scipy            | 1.7.3  | Пакет із статистичними функціями та розподілами      |
| keras            | 2.4.0  | Шари, функції витрат, активації для нейронної мережі |
| tensorflow       | 2.4.1  | Побудова, навчання та використання нейронної мережі  |
| matplotlib       | 3.5.0  | Інструменти для візуалізації та анімації             |
| random           | 3.1.0  | Рандомізація фейкових даних для змагальної мережі    |

### 3.3. Набори вхідних даних

Для того, щоб повноцінно охопити задачу оптимального транспортування, виберемо три різноманітних набори даних, між якими будемо шукати найоптимальніше відображення.

Перший набір належить до класичної логістичної задачі — набір з точок виробництва та набір з точок збуту. В доповнення до усього задача є незбалансованою, так як кількість продукції на пекарнях більша, ніж її попит в магазинах. На рисунку 3.3 зображена мапа з точками заводів (помаранчеві) та магазинів (сині).

Сумарний попит в магазина дорівнює 378 одиниць. Натомість заводи сумарно виробляють 793 одиниці.

Евклідова відстань між точками показана на рисунку 3.4, де жовтим виділено найдальші точки, а синім найближчі. Також ця матриця є матрицею вартості для логістичних задач. Для коректності обрахунків її варто нормалізувати.

Heatmap visualization showing the relationship between Shop (X-axis, 0, 2, 4) and Factory (Y-axis, 0, 1, 2, 3, 4, 5, 6, 7). The color scale ranges from 0 to 200,000. The heatmap shows varying values across the grid, with a color bar on the right indicating the magnitude.

Рисунок 3.4 - Матриця вартості

Генератор, що буде навчатись а цих даних — найпростіший з усіх трьох та показаний на рисунку 3.5.

Model: "Generator"

| Layer (type)              | Output Shape | Param # |
|---------------------------|--------------|---------|
| dense_1 (Dense)           | (None, 5)    | 45      |
| leaky_re_lu_1 (LeakyReLU) | (None, 5)    | 0       |
| Total params: 45          |              |         |
| Trainable params: 45      |              |         |
| Non-trainable params: 0   |              |         |

Рисунок 3.5 - Генератор та дискримінатор для логістичної задачі

Друга група наборів належить до проблеми трансферного навчання. Перший та другий набори належать до відкритого модулю `tf.keras.datasets` [21] — CIFAR10, набір з картинок для їх класифікації з 10 різноманітними класами та набір Fashion MNIST з картинками одягу. З другого набору видалимо співвідношення між елементами та лейблами класу, залишивши їх окремо та знайдемо відображення одного класу в інший.

Ця задача також належить до незбалансованої проте потребує попередньої обробки даних через різну розмірність. Розмірність набору з тваринами/ транспортом — (5000, 32, 32, 3). Набір з одягом має розмірність (6000, 28, 28). Тому на генератор лягає додаткова задача генерації кольору. Його архітектура в даному контексті має вигляд як на рисунку 3.6.

Model: "Generator\_image"

| Layer (type)              | Output Shape             | Param # |
|---------------------------|--------------------------|---------|
| dense_2 (Dense)           | (None, 6000, 28, 640)    | 18560   |
| leaky_re_lu_2 (LeakyReLU) | (None, 6000, 28, 640)    | 0       |
| dropout_1 (Dropout)       | (None, 6000, 28, 640)    | 0       |
| reshape_1 (Reshape)       | (None, 5000, 32, 32, 21) | 0       |
| dense_3 (Dense)           | (None, 5000, 32, 32, 3)  | 66      |
| leaky_re_lu_3 (LeakyReLU) | (None, 5000, 32, 32, 3)  | 0       |
| Total params: 18,626      |                          |         |
| Trainable params: 18,626  |                          |         |
| Non-trainable params: 0   |                          |         |

Рисунок 3.6 - Генератор для набору картинок

Третій тип задачі — заміна кольорової гама на картинці іншою. Дані натеper збалансовані. Одна картинка чорно-біла, на яку ми будемо додавати кольорову гаму з іншої картинки. На рисунках 3.7 та 3.8 представлені тестові малюнки та їх кольорова гама.



Рисунок 3.7 - Картинки для перенесення кольору

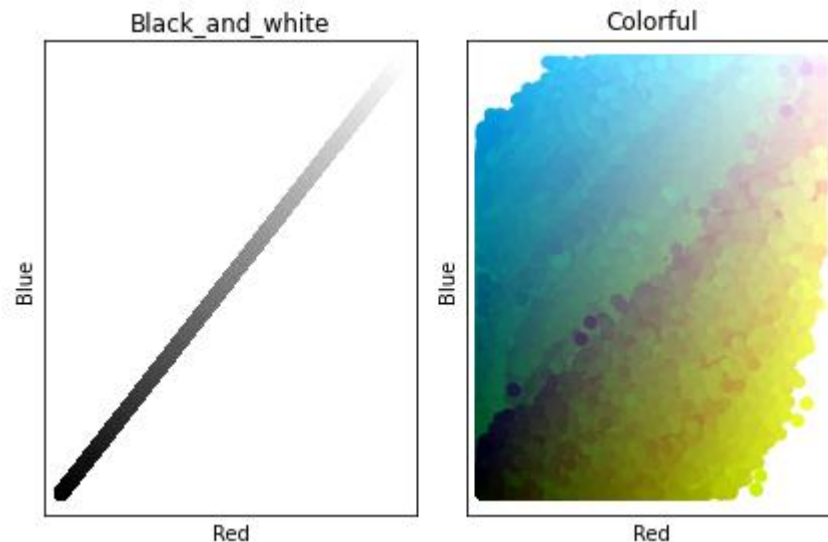


Рисунок 3.8 - Кольорова гама обох картинок

Ця різноманітність даних, та можливість їх обробки методом машинного навчання відповідає на питання актуальності та необхідності вирішення задачі.

### 3.4. Аналіз результатів

Для першого набору логістичних даних перш за все необхідно навчити дискримінатор на реальних даних, призначивши їм лейблу 1. А також створити «фейкові» дані з лейблою 0 для навчання дискримінатора.

Функція витрат для конкретно цієї задачі визначена як лінійна оболонка евклідової відстані між згенерованими даними та реальними і матрицею витрат помноженою на прапор передачі даних між точками.

У генератора функцією активації є  $\text{ReLU} = \max(0, z)$  для запобігання потрапляння від'ємних значень до вектору з вагами. Проте найбільш наближеним до реальної ситуації є використання  $\text{LeakyRelu} = \max(-\alpha z, z)$ ,  $\forall \alpha > 0$ , як функції активації, розглядаючи ситуацію, коли

товар з магазинів може повертатись на фабрики та з них перевозитись до магазинів, створюючи неklasичний розв'язок транспортної задачі.

Точність та швидкість знаходження оптимального відображення залежить від методу оптимізації функції витрат, що показано в таблиці 3.1.

Таблиця 3.1 — Залежність між функцією втрат та методом оптимізації

| Метод оптимізації        | К-сть кроків до виходу | root_mean squared_error | З-ня цільової ф-ії |
|--------------------------|------------------------|-------------------------|--------------------|
| Adagrad                  | 1167                   | 0.2859                  | 366545.66          |
| SGD momentum             | 157                    | 2.8097                  | 375852.0           |
| GradientDescentOptimizer | 145                    | 9.8714                  | 270722             |
| Adam                     | 326                    | 0.4331                  | 203584             |

Для даної задачі найкращі результати за точністю відображення та мінімізацією матриці витрат показав метод Адама – метод стохастичного градієнтного спуску, оснований на адаптивній оцінці моментів першого та другого порядків.

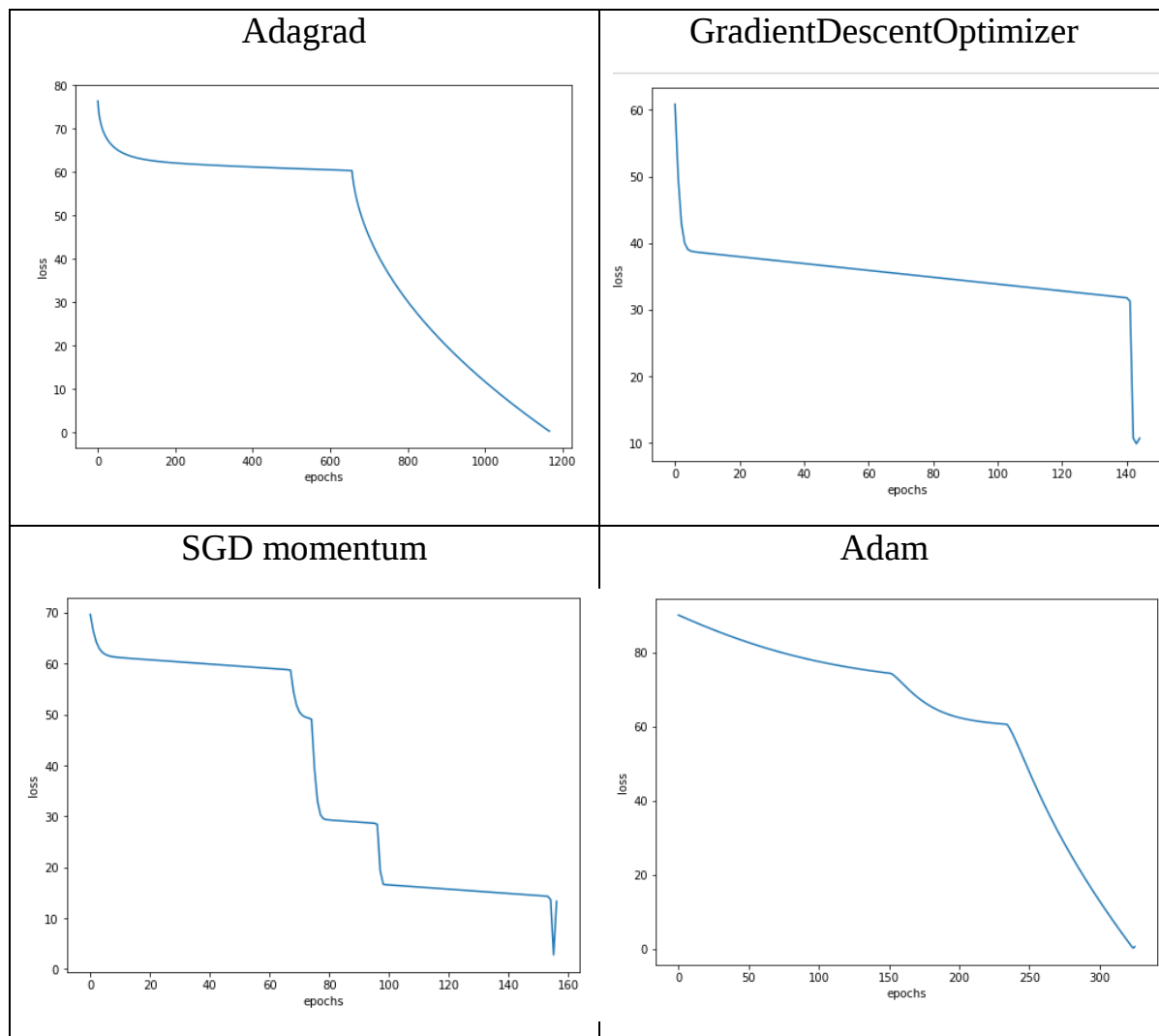
У таблиці 3.2 показані графіки залежності функції витрат від епохи навчання. Звертаючи увагу на плавність та швидкість досягнення мінімальної точки можна підтвердити коректність вибору методу оптимізації.

Другий набір даних має складність у різній розмірності вхідних та вихідних даних, додатковій необхідності генерації кольору та незбалансованості даних. Розмір наборів є набагато більшим за першу задачу, а швидкість навчання напряду залежить від розміру.

Функцією витрат для даної задачі є різниця між цільовим та згенерованим набором, а також сила перетворення даних, що

визначається як міра відхилення матриці вагів від одиниці – ідеального стану, коли дані абсолютно не змінюються.

Таблиця 3.2 — Залежність між функцією втрат та епохою



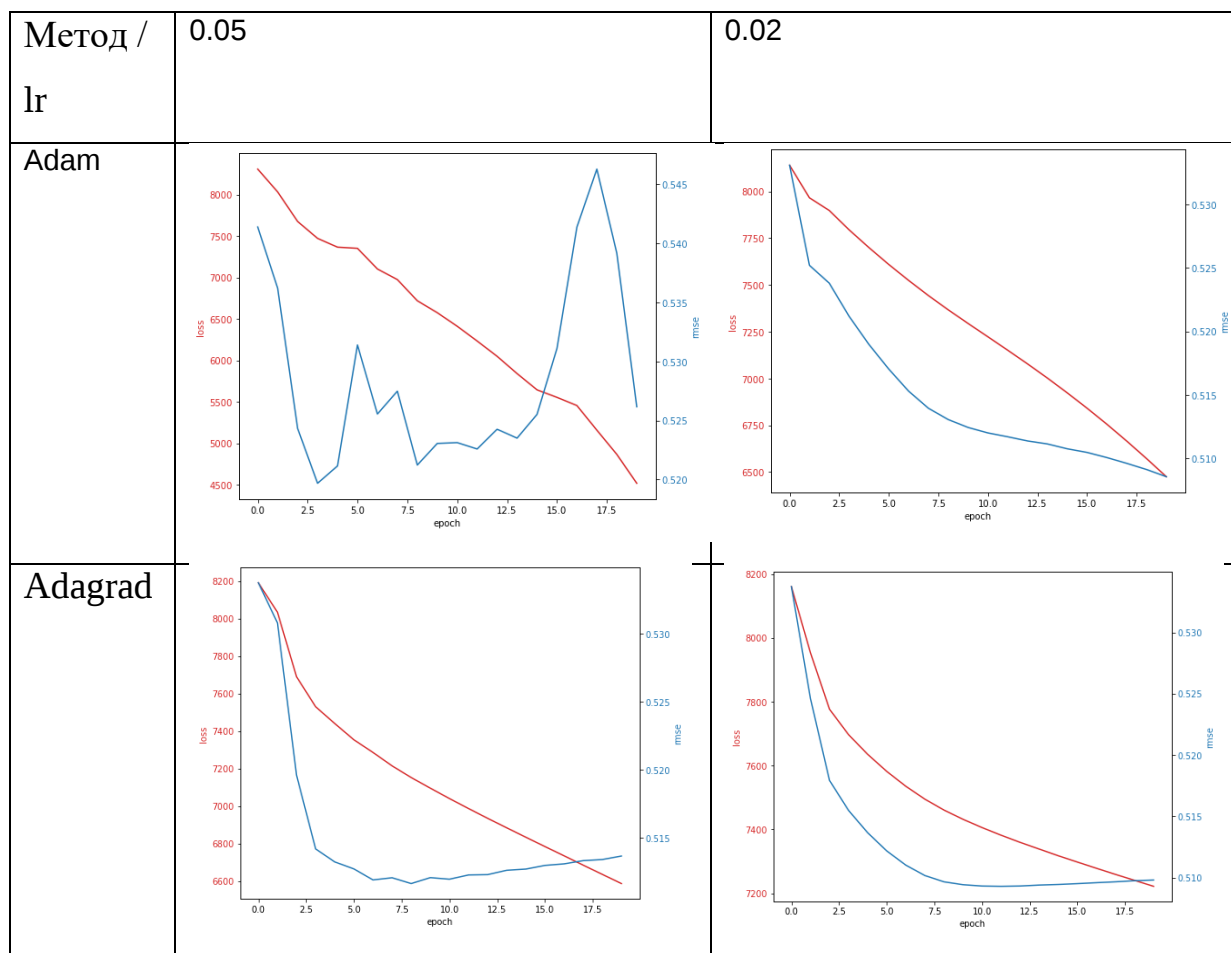
Було проведено декілька експериментів з глобальним параметром швидкості оптимізації learning rate та найкращі результати визначені у таблиці 3.3.

Таблиця 3.3 — Залежність між функцією втрат та методом оптимізації

| Метод оптимізації | learning rate | root_mean squared_error | З-ня цільової ф-її |
|-------------------|---------------|-------------------------|--------------------|
| Adagrad           | 0.05          | 0.514                   | 6602.47            |
| Adagrad           | 0.02          | 0.509                   | 7221.9             |
| Adam              | 0.05          | 0.526                   | 4523.11            |
| Adam              | 0.02          | 0.508                   | 6510.48            |
| RMSprop, moment   | 0.005         | 0.510                   | 7132.56            |

Графіки навчання мережі на різних оптимізаційних методах показано у таблиці 3.4.

Таблиця 3.4 — Залежність між функцією втрат та епохою



Судячи з результатів навчання найкращі значення показав метод Адама, проте необхідно варіювати між значенням `learning rate` щоб не натрапити на проблему перенавчання.

Плавність навчання та постійне спадання `rmse` вказує на те, що найкращою стратегією є використання методу Адама з коефіцієнтом навчання 0.02.

Третій варіант даних є набором з двох картинок між якими ми переносимо кольори таким методом, щоб відображення було мінімальним.

Результатом використання кольору з кольорової картинки на чорно-білій є рисунок 3.9.



Рисунок 3.8 - Результат оптимального транспортування кольору на чорно-білу картинку

Присутність на картинці тьмяних та однотонних кольорів є очевидним, адже мінімізуються затрати перенесення і серед усіх присутніх кольорів найближчими виявились темно-зелений та світло-зелений, що найбільш схожі на чорно-білу гаму.

Для оптимізації використовувався все той же метод стохастичного градієнта Адама. З даною функцією витрат в усіх трьох випадках

найкращі результати показував саме цей метод, тому можна прийняти, що для подальших експериментів оптимально використовувати саме його.

Проводячи другий експеримент з перенесення кольору, вже між двома кольоровими картинка. Вони самі та їх кольорова гама показана на рисунку 3.9

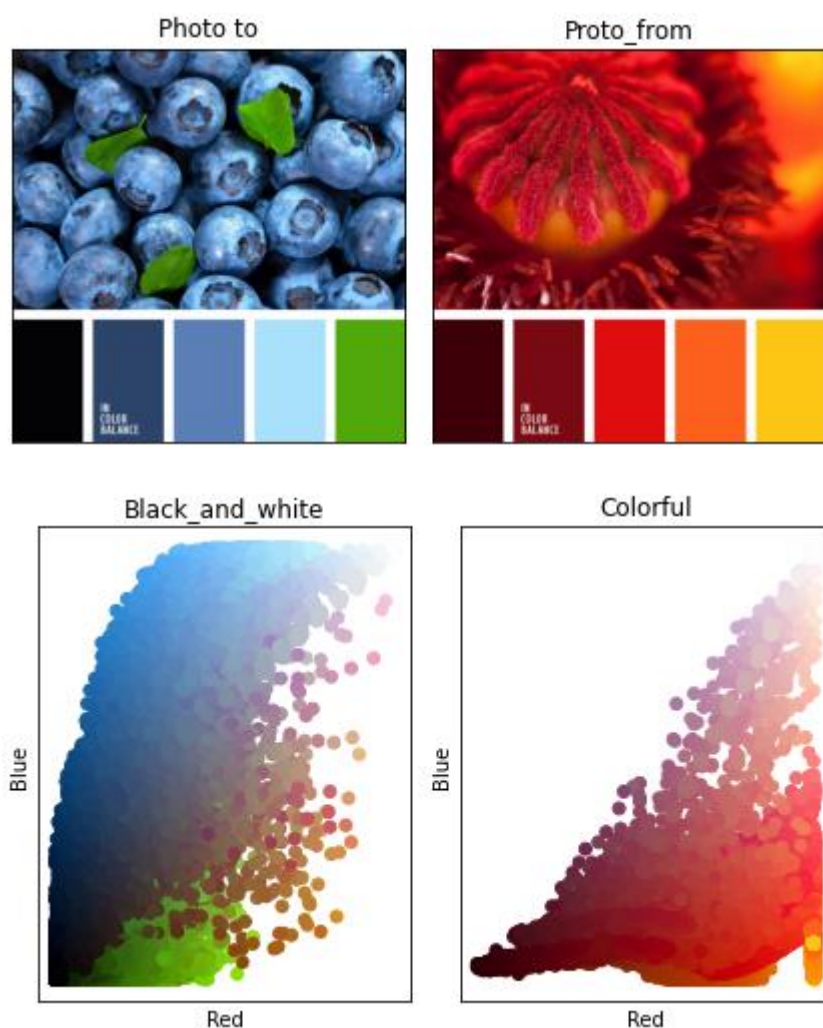


Рисунок 3.9 – Картинки та їх кольорова гама для другого експерименту

Наразі для навчання були використані батчі через достатньо великий набір даних – близько 500 тисяч даних. Також, існує ще одна відмінність між минулими моделями – через збалансованість задачі ми

можемо для навчання використовувати не всю вибірку одночасно, а більш стандартно, по-елементно.

Після достатньо швидкого навчання мережі – знадобилось близько 3 епох – отримуємо зображення на рисунку 3.10.

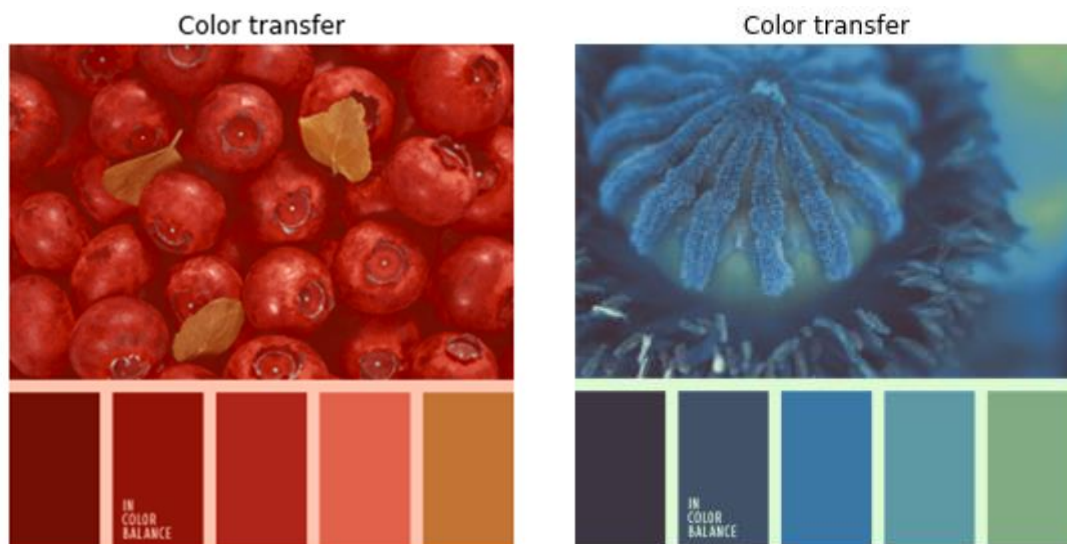


Рисунок 3.10 - Результат оптимального транспортування кольору на кольорову картинку

При необхідності – існує можливість побудувати більш складні зв'язки між шарами, дозволяючи розв'язувати складніші задачі з більш різноманітними даними.

### 3.5. Висновки до розділу

В третьому розділі було розглянуто практичне використання нейронної мережі для розв'язання задачі оптимального транспортування.

На прикладі трьох типів задач експериментальним шляхом було виявлено найкращу функцію витрат та метод оптимізації, що є

найкращим для використання в мережі. Загальна архітектура мережі для знаходження оптимального відображення є відносно простою – невелика кількість параметрів для навчання та шарів.

Зважаючи на складність задачі, розмірність даних, масштаб та збалансованість, архітектуру нейронної мережі можна змінювати, додавати до неї необхідні шари, функції активації, змінювати функцію витрат, чого неможливо зробити з класичними методами знаходження оптимального відображення.

На всіх трьох прикладах задач оптимального транспортування розв'язок був знайдений відносно швидко та достатньо точно.

## РОЗДІЛ 4 РОЗРОБКА СТАРТАП ПРОЕКТУ

### 4.1. Опис ідеї проекту

Метою магістерської дисертації є дослідження використання методів машинного навчання, зокрема генеративної змагальної мережі, для вирішення задачі оптимального транспортування на наборах з різних сфер використання. Зміст ідеї, потенційні напрямки застосування та головні вигоди, що може отримати користувач товару представлено у таблиці 4.1

Таблиця 4.1 — Опис ідеї стартап-проекту

| Зміст ідеї                                                                                           | Напрямки застосування                                                                  | Вимоги для користувача                                                                                                           |
|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Побудова та навчання генеративної нейронної мережі для вирішення задачі оптимального транспортування | Дослідницькі сфери — застосування у трансферному навчанні, пошуку схожості розподілів. | Швидкодія, без необхідного лейблуння та попередньої обробки. Можливість розв’язування незбалансованих та великомасштабних задач. |
|                                                                                                      | Практичні сфери — компанії, ігри, кінопроекти.                                         |                                                                                                                                  |

Запропонована модель відрізняється від існуючих методів своєю універсальністю та простотою. Було визначено перелік техніко – економічних властивостей та характеристик ідеї. Наступним етапом проведено аналіз потенційно можливих техніко-економічних властивостей та характеристик ідеї та існуючих аналогів. Для проведення такого аналізу, визначено попереднє коло опонентів, що

представлені на ринку. Результати досліджень представлено у таблиці 4.2.

Таблиця 4.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

| Техніко-економічні характеристики ідеї | Типи методів |                      |                      | Слабка сторона                   | Ней-на сторона                               | Сильна сторона                                |
|----------------------------------------|--------------|----------------------|----------------------|----------------------------------|----------------------------------------------|-----------------------------------------------|
|                                        | Мій проєкт   | Пошук опорного плану | Знаходження відстані |                                  |                                              |                                               |
| Масштаб та розмір даних                | +            | -                    | +                    |                                  |                                              | Не залежить від розмірності та масштабу даних |
| Простота знаходження розв'язку         | +            | +                    | -                    |                                  | Простота відносно юзера, що має набори даних |                                               |
| Швидкодія методу                       | +-           | -                    | +                    | Навчання мережі може бути довшим |                                              |                                               |

Продовження таблиці 4.2

| Техніко-<br>економічні<br>характерист<br>ики ідеї | Типи методів      |                                |                             | Слабка<br>сторона                                             | Ней-на<br>сторона | Сильна<br>сторона |
|---------------------------------------------------|-------------------|--------------------------------|-----------------------------|---------------------------------------------------------------|-------------------|-------------------|
|                                                   | Мій<br>про<br>ект | Пошук<br>опорно<br>го<br>плану | Знаходже<br>ння<br>відстані |                                                               |                   |                   |
| Швидкодія<br>методу                               | + -               | -                              | +                           | за<br>знаходже<br>ння<br>відстані<br>методом<br>Грінхорн<br>у |                   |                   |

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

#### 4.2. Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

У таблиці 4.3 приведено показники технічного аудиту ідеї проекту, та визначено за якою технологією буде виготовлено продукт, проведено аналіз існування необхідних технологій та їх доступність авторам проекту.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

| № | Ідея проекту                            | Технології її реалізації | Наявність технологій | Доступність технологій                                    |
|---|-----------------------------------------|--------------------------|----------------------|-----------------------------------------------------------|
| 1 | Розробка нейронної сітки                | Python                   | +                    | +                                                         |
| 2 | Бібліотека для побудови нейронних мереж | Tensorflow               | +                    | +                                                         |
| 3 | Сервер для хмарних обчислень            | Google Cloud Platform    | +                    | + (достатньо дорого, проте є програма підтримки проектів) |
| 4 | Сервер для хмарних обчислень            | Amazon Web Services EC2  | +                    | -                                                         |
| 5 | UI проекту                              | Python                   | +                    | + (достатньо довгий час написання)                        |
| 6 | UI проекту                              | Tableau                  | +                    | - (70 \$ за юзера в місяць)                               |

Обрані технології Python+Tensorflow+GCP.

За результатами аналізу таблиці робиться висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким це доцільно зробити (з поміж названих

технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

#### 4.3. Аналіз ринкових можливостей запуску стартап-проекту

Визначено ринкові можливості, які можна використати під час ринкового впровадження проекту та ринкових загроз, які можуть зашкодити реалізації проекту. В таблиці 4.4 показано результати аналізу попиту та стану справ на ринку.

Таблиця 4.4 - Результати аналізу попиту та стану справ на ринку

| № | Показники стану ринку                                    | Характеристика |
|---|----------------------------------------------------------|----------------|
| 1 | Кількість головних гравців, од                           | До 5           |
| 2 | Динаміка ринку (якісна оцінка)                           | Зростає        |
| 3 | Наявність обмежень для входу (вказати характер обмежень) | Немає          |
| 4 | Специфічні вимоги до стандартизації та сертифікації      | Немає          |
| 5 | Середня норма рентабельності в галузі (або по ринку), %  | 45%            |

Роблячи аналіз отриманих результатів, можна сказати, що ринок є досить привабливим для входження за попереднім оцінюванням.

Наступним етапом є визначення потенційних груп клієнтів, їх характеристики та формування орієнтовного переліку вимог до товару для кожної групи. Результати досліджень представлено у таблиці 4.5.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

| № | Потреба, що формує ринок | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару      |
|---|--------------------------|--------------------------------------------|-------------------------------------------------------------------|----------------------------------|
| 1 |                          | Логістичні підприємства                    | Бажання підтримки конкурентоспроможності                          | Низька ціна та висока швидкодія  |
| 2 |                          | Наукові стартапи                           | Використання нових способів обробки даних                         | Швидкодія, простота використання |
| 3 |                          | Кіноіндустрія                              | Нові і швидкі методи обробки відео та зображень                   | Низька ціна, висока швидкодія    |
| 4 |                          | Медицина                                   | Швидкі і точна методи знаходження міри схожості двох картинок     | Точність та швидкодія            |

Далі, проводимо аналіз ринкового середовища та складаємо таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають. Результати представлено у таблиці 4.6 та 4.7. Фактори в таблиці подаються в порядку зменшення значущості.

Таблиця 4.6 – Фактори загроз

| №<br>п/п | Фактор                      | Зміст загрози                                                                              | Можлива реакція<br>компанії                  |
|----------|-----------------------------|--------------------------------------------------------------------------------------------|----------------------------------------------|
| 1        | Поява більш швидких методів | Можлива поява та велика популярність нового методу, що матиме кращі характеристики         | Постійне покращення та контроль ринку        |
| 2        | Відсутність інвестицій      | Розробка проекту значно ускладниться, так як прийдеться працювати на ентузіазмі та продажі | Пошук джерел інвестицій                      |
| 3        | Конкуренція                 | Збільшення зацікавленості основних гравців в цій галузі                                    | Постійне покращення UI та пошук нових рішень |

Таблиця 4.7 – Фактори можливостей

| №<br>п/п | Фактор                      | Зміст можливості                         | Можлива<br>реакція компанії      |
|----------|-----------------------------|------------------------------------------|----------------------------------|
| 1        | Інвестиції                  | Отримання коштів на проект               | Вкладання інвестицій в розробку  |
| 2        | Отримання контракту         | Підписання контракту з великою компанією | Збільшення обсягів виробництва   |
| 3        | Покращення існуючого методу | Збільшення конкурентоспроможності        | Збільшення популярності продукту |

Наступним етапом проведемо аналіз пропозиції та визначаємо рівень конкуренції. Результати аналізу представлено у таблиці 4.8.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

| № | Особливості Конкурентного середовища | В чому проявляється дана характеристика                     | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|---|--------------------------------------|-------------------------------------------------------------|----------------------------------------------------------------------------------------|
| 1 | Чиста конкуренція                    | Існує багато відкритих алгоритмів та бібліотек              | Розвиток технологічності, використання переваги у гнучкості.                           |
| 2 | Міжнародна конкуренція               | Транспортні задачі не прив'язані до географічної місцевості | Налагодження зав'язків в усіх країнах.<br>Використання декількох мов в UI.             |
| 3 | Міжгалузева конкуренція              | Транспортна задача використовується у багатьох галузях      | Можливість вести бізнес в ширшому діапазоні галузей                                    |
| 4 | Товарно-родова конкуренція           | Створення товарів однієї категорії                          | Ведення активної рекламної кампанії                                                    |
| 5 | Конкурентні переваги нецінові        | Покупці готові платити більше за більш якісну систему       | Спрямування більшої частини доходів на розвиток технології                             |
| 6 | Марочна конкуренція                  | Система призначена для спільної цільової аудиторії          | Необхідно зосередити увагу клієнта на бренді та унікальності                           |

Аналіз конкуренції в галузі за моделлю 5 сил М. Портера приведено у таблиці 4.9.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

|                  |                                                                            |                                   |               |         |                                         |
|------------------|----------------------------------------------------------------------------|-----------------------------------|---------------|---------|-----------------------------------------|
| Складові аналізу | Прямі конкуренти в галузі                                                  | Потенційні конкуренти             | Постачальники | Клієнти | Товари-замінники                        |
|                  | Компанії в різних галузях, що використовують інший спосіб вирішення задачі | Немає                             | Немає         | Немає   | Мат. методи                             |
| Висновки         | Конкурентна боротьба середньої складності проте в різних галузях           | Наявна можливість виходу на ринок | Ні            | Ні      | Товари-замінники мають зовсім інший вид |

За результатами аналізу таблиці робиться висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію. З огляду на конкуренцію, робота на ринку реальна та можлива. Конкуренти є, проте вони знаходяться кожен у своїй галузі.

Визначимо фактори конкурентоспроможності, їх наведено у таблиці 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

| №<br>п/п | Фактор<br>конкурентоспроможності | Обґрунтування                                                              |
|----------|----------------------------------|----------------------------------------------------------------------------|
| 1        | Низька собівартість              | Повторне використання вже визначеної архітектури для схожих типів задач    |
| 2        | Інноваційність                   | Нові та актуальні методи завжди приваблює більшість компаній та інвесторів |

За визначеними факторами конкурентоспроможності проводимо аналіз сильних та слабких сторін стартап-проекту. Порівняльний аналіз представлено у таблиці 4.11

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

| №<br>п/п | Фактор<br>конкурентоспроможності | Бали<br>1-20 | Рейтинг товарів-конкурентів у порівнянні з власним проектом |    |    |   |    |    |    |
|----------|----------------------------------|--------------|-------------------------------------------------------------|----|----|---|----|----|----|
|          |                                  |              | -3                                                          | -2 | -1 | 0 | +1 | +2 | +3 |
| 1.       | Міжгалузовість                   | 20           |                                                             |    |    |   | +  |    |    |
| 2.       | Інноваційність                   | 10           |                                                             |    |    | + |    |    |    |
| 3.       | Низька собівартість              | 5            |                                                             |    | +  |   |    |    |    |
| 4.       | Простота експлуатації            | 15           |                                                             |    |    |   |    |    | +  |

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11).

Таблиця 4.12 – SWOT – аналіз стартап-проекту

|                                                                      |                                                                                                      |
|----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Сильні сторони: досить висока точність, низька ціна, міжгалузевість. | Слабкі сторони: нижча швидкодія ніж у конкурентів, певна індивідуальність у підході до кожної задачі |
| Побудова на будь-яких даних.                                         | Використання тільки на нестандартних задачах.                                                        |

На основі SWOT – аналізу розроблено альтернативи ринкової поведінки для виведення стартап – проекту на ринок та орієнтовний час їх ринкової реалізації. Розроблені альтернативи показано у таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

| № п/п | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|-------|----------------------------------------------------------------|--------------------------------|-------------------|
| 1.    | Підписання контракту з великою компанією                       | Малоймовірно                   | 3-4 місяці        |
| 2.    | Вихід на ринок в якості потенційного проекту                   | Ймовірно                       | 3 місяці          |
| 3.    | Перехід на безкоштовне розповсюдження                          | Ймовірно                       | 4 місяці          |

Аналізуючи наявні перспективи найвигіднішим варіантом є реалізація продукту в кооперації з компаніями.

#### 4.4. Розроблення ринкової стратегії проекту

Розробка ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів. у. Опис цільових груп представлено у таблиці 4.14.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

| №<br>п/п | Опис<br>профілю<br>цільової<br>групи<br>потенційних<br>клієнтів | Готовність<br>споживачі<br>в<br>сприйняти<br>продукт | Орієнтовний<br>попит в<br>межах<br>цільової<br>групи<br>(сегменту) | Інтенсивність<br>конкуренції в<br>сегменті | Простота<br>входу у<br>сегмент |
|----------|-----------------------------------------------------------------|------------------------------------------------------|--------------------------------------------------------------------|--------------------------------------------|--------------------------------|
| 1.       | Логістичні<br>підприємства                                      | Висока                                               | 80%                                                                | Висока                                     | Висока                         |
| 2.       | Наукові<br>стартапи                                             | Висока                                               | 70%                                                                | Висока                                     | Висока                         |
| 3.       | Кіноіндустрія                                                   | Низька                                               | 45%                                                                | Низька                                     | Низька                         |
| 4.       | Медицина                                                        | Середня                                              | 25%                                                                | Низька                                     | Низька                         |

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Для роботи в обраному сегменті ринку сформулюємо базову стратегію розвитку. Визначення базової стратегії розвитку представлено в таблиці 4.15

Таблиця 4.15 – Визначення базової стратегії розвитку

| №<br>п/п | Обрана<br>альтернатива<br>розвитку проекту                    | Ключові<br>конкурентоспроможні<br>позиції відповідно до<br>обраної альтернативи  | Базова<br>стратегія<br>розвитку |
|----------|---------------------------------------------------------------|----------------------------------------------------------------------------------|---------------------------------|
| 1.       | Вихід на ринок в<br>якості<br>співробітництва з<br>компаніями | Можливість виходу<br>на ринок без<br>створення<br>фізичного готового<br>продукту | Стратегія<br>диференціації      |

У таблиці 4.16 представлено визначення базової стратегії конкурентної поведінки.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

| №<br>п/п | Чи є проект<br>«першопрохідцем»<br>на ринку? | Чи буде<br>компанія<br>шукати<br>нових<br>споживачів,<br>або забирати<br>існуючих у<br>конкурентів? | Чи буде<br>компанія<br>копіювати<br>основні<br>характеристики<br>товару<br>конкурента, і<br>які? | Стратегія<br>конкурентної<br>поведінки       |
|----------|----------------------------------------------|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------|
| 1.       | Ні                                           | 50/50                                                                                               | За потреби                                                                                       | Стратегія<br>заняття<br>конкурентної<br>ніші |

Визначення стратегії позиціювання представлено у таблиці 4.17.

Таблиця 4.17 – Визначення стратегії позиціювання

| №<br>п/п | Вимоги до<br>товару<br>цільової<br>аудиторії                                                | Базова<br>стратегія<br>розвитку | Ключові<br>конкурентоспроможні<br>позиції<br>власного<br>стартап-проекту     | Вибір<br>асоціацій,<br>які мають<br>сформувати<br>комплексну<br>позицію<br>власного<br>проекту (три<br>ключових) |
|----------|---------------------------------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| 1.       | Висока<br>точність,<br>різноманітність<br>галузей<br>використання,<br>достатня<br>швидкодія | Стратегія<br>диференціа<br>ції  | Інноваційність,<br>новітність,<br>різноманітність,<br>універсальність даних. | Адаптивність,<br>універсальніс<br>ть,<br>швидкодія,<br>точність.                                                 |

#### 4.5. Розроблення маркетингової програми стартап-проекту

На наступному кроці визначимо цінові межі, якими необхідно керуватися при встановленні ціни на потенційний товар. Визначення меж встановлення ціни показано у таблиці 4.18.

Таблиця 4.18 – Визначення меж встановлення ціни

| №<br>п/п | Рівень цін на<br>товари замітники | Рівень цін<br>на<br>товари-<br>аналоги | Рівень доходів<br>цільової групи<br>споживачів | Верхня та<br>нижня межі<br>встановлення<br>ціни на<br>товар/послугу |
|----------|-----------------------------------|----------------------------------------|------------------------------------------------|---------------------------------------------------------------------|
| 1.       | Відсутні                          | Невідомо                               | 4000 \$/місяць                                 | 2000 – 5000 \$                                                      |

Формування системи збуту представлено у таблиці 4.19.

Таблиця 4.19 – Формування системи збуту

| №<br>п/п | Специфіка<br>закупівельної<br>поведінки цільових<br>клієнтів | Функції<br>збуту, які<br>має<br>виконувати<br>постачальник<br>товару | Глибина<br>каналу<br>збуту | Оптимальна<br>система<br>збуту |
|----------|--------------------------------------------------------------|----------------------------------------------------------------------|----------------------------|--------------------------------|
| 1.       | Договір з компанією, в<br>яку інтегрується<br>продукт        | Інтеграція в<br>систему<br>клієнта                                   | Один рівень                | Комбінована                    |

Розробимо концепцію маркетингової комунікації. Її розробку представлено у таблиці 4.20.

Таблиця 4.20 – Концепція маркетингових комунікацій

| Специфіка поведінки цільових клієнтів | Канали комунікацій | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення           | Концепція рекламного звернення |
|---------------------------------------|--------------------|--------------------------------------------|--------------------------------------------|--------------------------------|
| Люди, що відкриті до нових методів    | Реклама на сайті   | Швидкодія, інноваційність, актуальність    | Демонстрація конкурентних переваг продукту | Приклади використання          |

#### 4.6. Висновок до розділу

В даному розділі проведено підготовчий аналіз для впровадження розробленої моделі в якості стартап-проекту. Досліджено аналогічні конкурентні моделі, встановлено сильні та слабкі сторони системи в порівнянні з ними.

Також було досліджено можливі шляхи розповсюдження продукту та його ймовірну аудиторію, рівень доходів та ймовірну ціну продукту, що розробляється.

Встановлено, що хоча модель, що розробляється, і є менш швидкою в порівнянні з наявними на ринку рішеннями, але через можливість використання на різногалузевих даних з різною складністю, модель потенційно має більш високу популярність, що теоретично може привезти до того, що проект зможе зайняти нішу, яка ще не містить значної конкуренції та посягати на співробітництво з великими компаніями.

## ВИСНОВКИ

В даній магістерській дисертації було досліджено можливість вирішення задач оптимального транспортування методами машинного навчання, а саме з використанням генеративної змагальної мережі.

В ході виконання роботи було проведено ознайомлення з актуальністю задачі оптимального транспортування та сферами її використання.

Дане дослідження наведено у першому розділі. Також було проаналізовано види транспортних задач, такі як незбалансовані або масштабовані.

В другому розділі було проаналізовано існуючі методи розв'язку задачі, виявлено їх переваги та недоліки. Також було висунуто метод машинного навчання, який використовувався для реалізації цілей встановлених даною магістерською дисертацією.

В третьому розділі було наведено архітектуру обраної системи, проведено її навчання на трьох різноманітних видах даних та проаналізовано результати. Загалом було встановлено, що створена система забезпечує досить високі результати в плані точності знаходження рішення, але може мати дещо нижчу швидкодію при великих наборах даних.

Знаходження та застосування універсальної функції витрат, що допоможе повноцінно описати всі можливі варіанти використання, належить до сфери майбутніх доопрацювань, та допоможе мати проекту ще більший потенціал.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Alvarez-Melis D., Fusi N. Measuring dataset similarity using optimal transport, 24 вересня 2020. URL: <https://www.microsoft.com/en-us/research/blog/measuring-dataset-similarity-using-optimal-transport/> (дата звернення 01.10.2021)
2. Schmidt M., Prof. Weickert J. Hauptseminar Optimal Transport in Image Processing. *Seminar Optimal Transport in Image Processing*. Winter Term 2015/16. URL: <https://www.mia.uni-saarland.de/Teaching/otip15.shtml> (дата звернення 01.10.2021)
3. Frigo O., Sabater N., Demoulin V., Hellier P. Transportation for Example-Guided Color Transfer. ACCV. 2014. URL: [http://oriel.github.io/color\\_transfer.html](http://oriel.github.io/color_transfer.html)
4. Grattan-Guinness I. Companion Encyclopedia of the History & Philosophy of the mathematical sciences. *JHU Press*. 2003. Vol. 1. P. 831.
5. Muzellec B., Josse J., Boyer C., Cuturi M. Missing Data Imputation using Optimal Transport. arXiv preprint arXiv:2002.03860. Jul. 2020. URL: <https://arxiv.org/abs/2002.03860>
6. Alvarez-Melis D., Jaakkola T. S. Gromov-Wasserstein Alignment of Word Embedding Spaces. arXiv preprint arXiv:1809.00013. Aug 2018. URL: <https://arxiv.org/pdf/1809.00013.pdf>
7. Богачов В. І., Колесніков А. В. Задача Монжа–Канторовича: достижения, связи и перспективы. УМН. 2012. Том 67, Вып. 5(407). С. 3–110. URL: <http://www.mathnet.ru/links/5ea337dab4c5e0341f69bc8e7286583c/rm9490.pdf>
8. Канторович Л. В. Математические методы организации и планирования производства. Ленинград: ЛГУ, 1939. 67 с.
9. Jin K., Liu C., Xia C. Two-sided Wasserstein Procrustes Analysis. Proceedings of the Thirtieth International Joint Conference on

Artificial Intelligence. 2021. P. 3515-3521. URL: <https://www.ijcai.org/proceedings/2021/0484.pdf>

10. Schrieber J. Algorithms for Optimal Transport and Wasserstein Distances, 2019. URL: <shorturl.at/nvBZ6>

11. Іксанов О.М., Шевченко В.І. Транспортна задача, її властивості та методи розв'язування. 2010. URL: <http://www.cyb.univ.kiev.ua/library/books/iksanov-26.pdf>

12. Aurenhammer F., Hoffmann F., Aronov B.. Minkowski-Type Theorems and Least-Squares Clustering. *Algorithmica*. 1998. Vol. 20. P. 61–76. URL: <https://link.springer.com/article/10.1007%2FPL00009187>

13. Kitagawa, J., Mérigot, Q., Thibert, B.: Convergence of a Newton algorithm for semi-discrete optimal transport. 2017. URL: <https://arxiv.org/abs/1603.05579>

14. Cuturi, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. *Proc. NIPS*. 2013. P. 2292–2300.

15. Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domain / Solomon, J. et. al. *ACM Trans. Graph*. 2015. Vol. 34, No. 4. P. 66:1–66:11.

16. Schmitzer, B. Stabilized Sparse Scaling Algorithms for Entropy Regularized Transport Problems. 2016. URL <https://arxiv.org/abs/1610.06519>

17. Altschuler, J., Weed, J., Rigollet, P. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. *CoRR abs/1705.09634* .2017. URL <https://arxiv.org/abs/1705.09634>

18. M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*. 2013. No. 26. P. 2292–2300.

19. Yang K. D., Uhler C. Scalable Unbalanced Optimal Transport using Generative Adversarial Networks. Aug 2019. arXiv preprint arXiv:1810.11447v2. URL: <https://arxiv.org/pdf/1810.11447.pdf>

20. Sato R., Yamada M., Kashima H. Fast Unbalanced Optimal Transport on a Tree. Jan 2021. arXiv preprint arXiv:2006.02703v3. URL: <https://arxiv.org/pdf/2006.02703.pdf>
21. Keras API reference / Datasets. URL: <https://keras.io/api/datasets/>

## ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

import tensorflow as tf
tf.config.run_functions_eagerly(True)
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from scipy.spatial import distance
from sklearn.preprocessing import normalize

from tensorflow.keras.layers import Input, Dense, Conv3DTranspose, LeakyReLU, ReLU,
Reshape, BatchNormalization, Activation, Conv3D
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.layers import Conv2D, Flatten, Dropout
from tensorflow.keras.optimizers import Adam

bakery_pos = data['bakery_pos']
bakery_prod = np.array([131., 98., 22., 80., 12., 148., 179., 123.])
cafe_pos = data['cafe_pos']
cafe_prod = np.array([52., 135., 32., 88., 71.])
Imap = data['Imap']
print('Factory production: {}'.format(bakery_prod))
print('Shop sale: {}'.format(cafe_prod))
print("Total pieces in shops : {}".format(cafe_prod.sum()))
print("Total pieces in factory : {}".format(bakery_prod.sum()))

fig = plt.figure(figsize=(12,10))
plt.imshow(data['Imap'], interpolation='nearest')
plt.scatter(cafe_pos[:,0],cafe_pos[:,1], s=cafe_prod)
plt.scatter(bakery_pos[:,0],bakery_pos[:,1], s=bakery_prod)
plt.show()

cost_mat = distance.cdist(bakery_pos, cafe_pos, 'sqeuclidean')
normed_matrix = normalize(cost_mat, axis=1, norm='l1')

def loss_log(y, y_):
    matr = tf.Variable(normed_matrix, dtype = tf.float32)
    tr = model_generator_1.weights[0]
    cost_err = tf.multiply(tr, matr)
    max_in_rows = tf.reduce_max(cost_err, 1)
    y_zero =tf.subtract(tf.size(y_),tf.math.count_nonzero(y_, dtype=tf.int32))*10
    return tf.cast(tf.sqrt(tf.losses.mean_squared_error(y,
y_))*(1+tf.reduce_sum(max_in_rows)),dtype=tf.float64)+ tf.cast(y_zero, dtype=tf.float64)

def getWeight():
    return model_generator_1.weights[0].numpy()

```

```

def generator_arcitech_log():
    generator = Sequential()
    generator.add(Input(shape = (8)))
    generator.add(Dense(5, name = 'weights'))
    generator.add(LeakyReLU(alpha=0.05))
    opt = tf.keras.optimizers.Adam(lr=0.0008)
    rmse=tf.keras.metrics.RootMeanSquaredError()
    generator.compile(loss=loss_log, optimizer = opt, metrics=[rmse])
    generator._name="Generator"
    return(generator)

model_generator_1 = generator_arcitech_log()

model_generator_1.summary()

def discriminator_arcitech_log():

    discriminator = Sequential()
    discriminator._name="Discriminator"

    discriminator.add(Input(shape = (5)))
    discriminator.add(Dense(32, activation='tanh'))
    discriminator.add(LeakyReLU(alpha=0.2))
    discriminator.add(Dropout(0.2))

    discriminator.add(Dense(16, activation='tanh'))
    discriminator.add(LeakyReLU(alpha=0.2))
    discriminator.add(Dropout(0.2))

    discriminator.add(Dense(4, activation='tanh'))
    discriminator.add(LeakyReLU(alpha=0.2))
    discriminator.add(Dropout(0.2))

    discriminator.add(Dense(1, activation='sigmoid'))

    opt = Adam(lr=0.0002 ,beta_1=0.5)
    discriminator.compile(loss='binary_crossentropy', optimizer= opt , metrics = ['accuracy'])

    return(discriminator)

model_discriminator_1 = discriminator_arcitech_log()
model_discriminator_1.summary()

def create_gan_log(discriminator, generator):
    discriminator.trainable=False
    gan = Sequential()
    gan.add(Input(shape = (8)))
    gan.add(generator)
    gan.add(discriminator)

```

```

gan._name = 'GAN'

opt = Adam(lr=0.0002,beta_1=0.5)
gan.compile(loss = loss_log, optimizer = opt)

return gan

gan_log = create_gan_log(model_discriminator_1,model_generator_1)
gan_log.summary()

def training(data_x, data_y, model_generator, epochs):
    history = model_generator.fit( data_x, data_y, epochs =epochs
                                   ,callbacks=tf.keras.callbacks.EarlyStopping(monitor='loss', min_delta
=0.00001)
                                   )
    return history

his = training(x_b, y_b, model_generator_1, 3000)
fig = plt.figure(figsize=(8,6))
plt.plot(his.history['loss'])
plt.xlabel('epochs')
plt.ylabel('loss')
plt.show()

matr = tf.Variable(cost_mat, dtype = tf.float32)
tr = model_generator_1.weights[0]
cost_err = tf.multiply(tr, matr)
max_in_rows = tf.reduce_max(cost_err, 1)
print(tf.reduce_sum(max_in_rows))

from tensorflow.keras.datasets import cifar10
from tensorflow.keras.datasets import fashion_mnist

def load_keras_dataset():
    (Xtrain, Ytrain), (_, _) = cifar10.load_data()
    (Xtest, Ytest), (_, _) = fashion_mnist.load_data()

    indice = np.where(Ytrain == 0)
    indice = indice[0]
    Xtrain = Xtrain[indice, :, :]
    Ytrain = Ytrain[indice]

    indice = np.where(Ytest == 0)
    indice = indice[0]
    Xtest = Xtest[indice, :, :]
    Ytest = Ytest[indice]

    # Normalization
    X = Xtrain.astype('float32')

```

```

X = (X - 127.5) / 127.5

X_2 = Xtest.astype('float32')
X_2 = (X_2 - 127.5) / 127.5

return X, X_2, Ytrain, Ytest

X_from, X_to, Y_from, Y_to = load_keras_dataset()
print(X_from.shape)
print(np.array([X_to]).shape)
print(Y_from.shape)

def loss_image(y, y_):
    wei = model_generator_2_image.weights
    weights1 = (tf.math.subtract(tf.math.abs(wei[0]),1))
    weights2 = (tf.math.subtract(tf.math.abs(wei[2]),1))
    y2=tf.math.abs(tf.reduce_sum(weights1))+tf.math.abs(tf.reduce_sum(weights2))
    return tf.sqrt(tf.losses.mean_squared_error(y, y_))*(1+y2)

def generator_arcitech_image():

    generator = Sequential()

    generator.add(Input(shape = (6000, 28, 28)))

    generator.add(Dense(640))
    generator.add(LeakyReLU(alpha=0.4))

    generator.add(Reshape((5000, 32, 32, -1)))

    generator.add(Dense(3))
    generator.add(LeakyReLU(alpha=0.4))

    rmse=tf.keras.metrics.RootMeanSquaredError()
    opt = Adam(lr=0.02,beta_1=0.5)
    generator.compile(loss = loss_image, optimizer = opt, metrics=[rmse])

    generator._name="Generator_image"
    return(generator)

model_generator_2_image = generator_arcitech_image()

model_generator_2_image.summary()

history = model_generator_2_image.fit( X_image, Y_image, epochs =20
                                     ,callbacks=tf.keras.callbacks.EarlyStopping(monitor='loss', min_delta
                                     =0.001)

```

```

    )
fig, ax1 = plt.subplots(figsize=(8,6))

color = 'tab:red'
ax1.set_xlabel('epoch')
ax1.set_ylabel('loss', color=color)
ax1.plot(history.history['loss'], color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'tab:blue'
ax2.set_ylabel('rmse', color=color) # we already handled the x-label with ax1
ax2.plot(history.history['root_mean_squared_error'], color=color)
ax2.tick_params(axis='y', labelcolor=color)

fig.tight_layout()
plt.show()

def im2mat(img):
    return img.reshape((img.shape[0] * img.shape[1], img.shape[2]))

def mat2im(X, shape):
    return X.reshape(shape)

I1 = pl.imread('bandw.jpg').astype(np.float64)/256
I2 = pl.imread('bf6e7333a8df6c470adcd308f97ed16.jpg').astype(np.float64)/256
I1= I1[:, :, :]
I2= I2[:, 855:, 570:, :]

def plot_gamma(from_ph, to_ph):
    plt.subplot(1, 2, 1)
    plt.scatter(from_ph[:, 0], from_ph[:, 2], c=from_ph)
    plt.xlabel('Red')
    plt.ylabel('Blue')
    plt.xticks([])
    plt.yticks([])
    plt.title('Black_and_white')

    plt.subplot(1, 2, 2)

    plt.scatter(to_ph[:, 0], to_ph[:, 2], c=to_ph)
    plt.xlabel('Red')
    plt.ylabel('Blue')
    plt.title('Colorful')
    plt.xticks([])
    plt.yticks([])
    plt.tight_layout()

```

```

plt.show()

def loss_color(y, y_):
    wei = model_generator_2_image.weights
    weights1 = (tf.math.subtract(tf.math.abs(wei[0]),1))
    y2=tf.math.abs(tf.reduce_sum(weights1))
    return (tf.losses.mean_squared_error(y, y_))* (1+y2)

def generator_arcitech_color():
    generator = Sequential()

    generator.add(Input(shape = (3)))

    generator.add(Dense(9))
    generator.add(BatchNormalization())
    generator.add(LeakyReLU(alpha=0.2))

    generator.add(Dense(3))
    generator.add(LeakyReLU(alpha=0.2))

    opt = tf.keras.optimizers.Adam(lr=0.005,beta_1=0.5)
    rmse=tf.keras.metrics.RootMeanSquaredError()
    generator.compile(loss=loss_color, optimizer = opt, metrics=[rmse])
    generator._name="Generator"
    return(generator)

model_generator_3 = generator_arcitech_color()
model_generator_3.summary()

history = model_generator_3.fit( face_to, face_from, epochs =10, batch_size=100
                                ,callbacks=tf.keras.callbacks.EarlyStopping(monitor='loss', min_delta
=0.0001)
                                )
transp_Xt_emd=model_generator_3.predict(face_to)
I2t = mat2im(transp_Xt_emd, I2.shape)

plt.figure()
plt.imshow(I2t)
plt.axis('off')
plt.title('Color transfer')
plt.tight_layout()
plt.show()

```