

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Комп'ютерних інтелектуальних технологій та систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Модель системи електронного навчання на основі
багатоагентного підходу.

Виконав:

студент 2 курсу, групи КІТм-21-1

Плеханов, Д.В.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні інтелектуальні
технології

Керівник проф., Безсонов О.О.

Допускається до захисту

(підпис)

Зав. кафедри

(підпис)

проф. Руденко О.Г.

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти другий (магістерський)
Спеціальність 123 – Комп'ютерна інженерія
Тип програми освітньо-професійна
Освітня програма Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Плеханову Денису Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Побудова реалістичних 3D зображень за допомогою штучної нейронної мережі NeRF

затверджена наказом по університету від “ _____ ” _____ 2022 р. № _____

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2022

3. Вхідні дані до роботи _____

1 Мова програмування Python

2 Технологія NeRF

4. Перелік питань, що потрібно опрацювати в роботі _____

1 Розкрити сутність поняття «штучні нейронні мережі» та головні складові мережі

2 Провести огляд нейронних мереж для створення зображень

3 Описати архітектуру нейронних мереж та математична сутність мережі NERF

4 Перелічити існуючі переваги та недоліки нейронної мережі NERF

5 Навести приклад практичного використання мережі NERF для створення зображень

6 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Слайд-презентація – 18 слайдів

Рисунки – 38

Таблиць – 2

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача та узгодження теми проекту	07.11.2022	виконано
2	Аналіз предметної області, постановка задачі	07.11.2022 – 11.11.2022	Виконано
3	Розробка алгоритмів і структури роботи	12.11.2022 – 18.11.2022	Виконано
4	Розробка структурної схеми нейропроцесора	19.11.2022 – 21.11.2022	Виконано
5	Реалізація інтелектуальної системи	22.11.2022 – 30.11.2022	Виконано
6	Розрахунок надійності розробленої системи	30.11.2022 – 01.12.2022	Виконано
7	Оформлення матеріалів роботи	02.12.2022 – 07.12.2022	Виконано
8	Подання її роботи керівникові	08.12.2022 – 19.12.2022	Виконано
9	Захист проекту	20.12.2022	виконано

Дата видачі завдання 07 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

професор Безсонов О.О
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка магістерської кваліфікаційної роботи: 90 с., 38 рис., 2 табл., 2 дод., 61 джерел.

НЕЙРОМЕРЕЖА, NERF, ПЕРЦЕПТРОН РОЗЕНБЛАТА, МАШИННЕ НАВЧАННЯ, 3D ЗОБРАЖЕННЯ, RESNET, ALEXNET.

Актуальність дослідження. Спектр областей впровадження нейронних мереж досить широкий в сучасному світі, в даній дослідницькій роботі була виділена одна з найбільш на сьогоднішній день популярних сфер їх застосування-медицина. Актуальність вибору цієї області також обумовлена і тим, що в серпні 2017 року були внесені в окремі законодавчі акти України з питань застосування інформаційно-телекомунікаційних технологій у сфері охорони здоров'я громадян та створення національних науково-медичних центрів, які легітимізують віддалені консультації та моніторинг пацієнтів. У роботі пропонується

- розкрити сутність поняття «штучні нейронні мережі» та головні складові мережі;
- розглянути особливості створення штучного нейрону, способи навчання та оптимізації;
- провести огляд нейронних мереж для створення зображень;
- описати архітектуру штучних нейронних мереж та математична сутність нейронної мережі NERF;
- перелічити існуючі переваги та недоліки нейронної мережі NERF;
- навести приклад практичного використання мережі NERF для створення зображень.

Метою кваліфікаційної роботи є проаналізувати особливості побудови реалістичних 3D зображень за допомогою штучної нейронної мережі NeRF.

Об'єктом дослідження є побудова реалістичних 3D зображень.

Предметом дослідження є сукупність необхідних умов, що забезпечують найкращий підхід до особливості побудови реалістичних 3D зображень за допомогою штучної нейронної мережі NeRF.

Рекомендації щодо можливостей застосування одержаних результатів.

ABSTRACT

Explanatory note of the qualification work: 90 pages, 38 figures, 2 tables, 2 appendices, 61 sources.

NEURAL NETWORK, NERF, ROSENBLATT PERCEPTRON, MACHINE LEARNING, 3D IMAGING, RESNET, ALEXNET.

The major goal of this thesis is

- reveal the essence of the concept of "artificial neural networks" and the main components of networks;

- consider the features of creating an artificial neuron, methods of training and optimization;

- review neural networks for creating images;

- describe the architecture of artificial neural networks and the mathematical essence of the NERF neural network;

- list the existing advantages and disadvantages of the NERF neural network;

- to give an example of the practical use of the NERF network to create images.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____

Кафедра _____ Комп'ютерних інтелектуальних технологій та систем _____

АНОТАЦІЯ

КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти другий (магістерський)

Побудова реалістичних 3D зображень за допомогою штучної
нейронної мережі NeRF

Виконав:

студент 2 курсу, групи КІТм-2х-х

Плеханов, Д.В

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні інтелектуальні
технології

Керівник посада, О.О , Безсонов

2022 р.

Актуальність теми дослідження полягає у тому що, системи розпізнавання графічної інформації відіграють важливу роль та є основними компонентами обчислювальних пристроїв, застосовуваних у різних сферах людської діяльності. Такі системи дозволяють значно спростити вирішення повсякденних задач, пов'язаних з обробкою графічної інформації, що й виділяє їм особливе місце у галузях. Наразі існує значна кількість нейромережевих систем здатних до використання у вирішенні задач розпізнавання об'єктів. Не зважаючи на значну кількість наукових досліджень що виконуються в цій області, наразі не існує найкращої в усіх аспектах моделі нейронної мережі. Однією із найперспективніших штучних нейронних мереж для створення 3D зображення є NeRF

Метою даної роботи: проаналізувати особливості побудови реалістичних 3D зображень за допомогою штучної нейронної мережі NeRF.

Об'єктом дослідження є побудова реалістичних 3D зображень.

Предметом дослідження сукупність необхідних умов, що забезпечують найкращий підхід до особливості побудови реалістичних 3D зображень за допомогою штучної нейронної мережі NeRF.

Методи дослідження: в роботі використані такі: пошуковий по наявній методичній та науковій літературі із аналізом знайденого матеріалу, порівняння, класифікація, проєктування, теоретичне моделювання, з'ясування причинно-наслідкових зв'язків, аналіз документації та результатів діяльності дослідників з проблеми проведеного дослідження.

Наукова новизна полягає у наявності оригінального дослідницького матеріалу по напрямку проведеного дослідження.

Практична цінність отриманих результатів полягає в наявності теоретичного матеріалу по дослідженню, відсіяного з-поміж іншого в процесі пошуку інформації по темі, та в систематизації матеріалу напрямку дослідження. Проведене дослідження має більш глибокий ступінь розробки напрямку дослідження, відносно попередніх досліджень вчених, дисертантів та дослідників напрямку дослідження.

У першому розділі розглянуто загальні характеристики нейронної мережі

У другому розділі була описана архітектура мережі NeRF

Третій розділ присвячено моделюванню мережі NeRF

Ключові слова: НЕЙРОМЕРЕЖА, NERF, ПЕРЦЕПТРОН РОЗЕНБЛАТА, МАШИННЕ НАВЧАННЯ, 3D ЗОБРАЖЕННЯ, RESNET, ALEXNET.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП.....	11
1.1 Сутність поняття «штучні нейронні мережі» та її головні складові	12
1.2 Особливості створення штучного нейрону, способи навчання та оптимізації.....	15
1.3 Огляд нейронних мереж для створення зображень.....	31
РОЗДІЛ 2 ОПИС ТА АРХІТЕКТУРА МЕРЕЖІ NERF ДЛЯ СТВОРЕННЯ 3D ЗОБРАЖЕНЬ.....	41
2.1 Архітектура штучних нейронних мереж	41
2.2 Математичний опис нейронної мережі NERF.....	43
РОЗДІЛ 3 ПРАКТИЧНЕ ВИКОРИСТАННЯ МЕРЕЖІ NERF ДЛЯ СТВОРЕННЯ 3D ЗОБРАЖЕНЬ.....	51
3.1 Існуючі переваги та недоліки нейронної мережі NERF.....	51
3.2 Практичне використання мережі NERF для створення зображень	59
ВИСНОВКИ.....	63
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	65
ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ АТЕСТАЦІЙНОЇ РОБОТИ	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
А.1 Різні типи нейронних структур	Ошибка! Закладка не определена.
А.2 Основні підходи до класифікації штучної нейронної мережі ...	Ошибка! Закладка не определена.
А.3 Приклади зображень створених за допомогою штучної нейронної мережі NERF	Ошибка! Закладка не определена.
ДОДАТОК Б КОДОВИЙ МАТЕРІАЛ АТЕСТАЦІЙНОЇ РОБОТИ	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.

Б.1 Модуль тренування штучного інтелекту, необхідного для машинного навчання **Ошибка! Закладка не определена.**

Б.2 Модуль контролю навчання штучного інтелекту **Ошибка! Закладка не определена.**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І
ТЕРМІНІВ

NeRF – нейронне поле випромінювання (англ., Neural Radiance Field)

ШНМ – штучна нейронна мережа

МН – машинне навчання

ANN – штучна нейронна мережа (анг. artificial neural network, ANN)

CNN – згорткові нейронні мережі (анг. convolutional neural network, CNN)

ВСТУП

У сучасному світі все більший інтерес проявляється в області вивчення нейронних мереж і нейрокомп'ютерів. Актуальність досліджень в цьому напрямку підтверджується великим числом публікацій, конференцій і різних практичних застосувань в галузях економіки, медицини, хімії, фізики та багатьох інших. За допомогою штучних нейронних мереж можна виконувати, наприклад, розпізнавання об'єктів на зображенні, передбачати коливання на ринку валют, моделювати модель прийняття рішень, визначати напрямки наукового пошуку, синтезувати мову по тексту, ставити діагнози і виявляти патології. На заході застосування штучних нейронних мереж (ШНМ) вже досить широко, в Росії ж це сприймається як щось екзотичне. Існували лише невелика кількість російських фірм, які використовують ШНМ в практичних цілях.

Графічні системи розпізнавання інформації відіграють важливу роль і є основним компонентом обчислювальних пристроїв, які використовуються в різних сферах людської діяльності. Такі системи можуть істотно спростити вирішення повсякденних завдань, пов'язаних з обробкою графічної інформації, що надає їм особливе місце в галузі. В даний час існує велика кількість нейромережових систем, які можна використовувати для вирішення задач розпізнавання об'єктів. Незважаючи на велику кількість наукових досліджень у цій галузі, наразі не існує єдиної моделі нейронної мережі, яка була б найкращою за всіма параметрами. Однією із найперспективніших штучних нейронних мереж для створення 3D зображення є NeRF.

1 РОЗДІЛ ЗАГАЛЬНА ХАРАКТЕРИСТИКА НЕЙРОННОЇ МЕРЕЖІ

1.1 Сутність поняття «штучні нейронні мережі» та головні складові мережі

Нейронна мережа, по суті, є математичною моделлю, яка імітує функціональні та структурні можливості природної нейронної мережі [1]. Основою кожної штучної нейронної мережі є штучний нейрон, проста математична функція (модель). Така модель має три простих набори правил: множення, додавання та активація. На вході штучного нейрона вхідні дані зважуються, тобто кожне вхідне значення множиться на окрему вагу.

У середині штучного нейрона знаходиться функція суми, яка поєднує всі зважені вхідні дані та зміщення. На виході штучного нейрона сума попередньо зважених вхідних даних і зміщень виникає через функцію активації (також відому як функція передачі). Схема штучного нейрона наведена на рисунку 1.1.

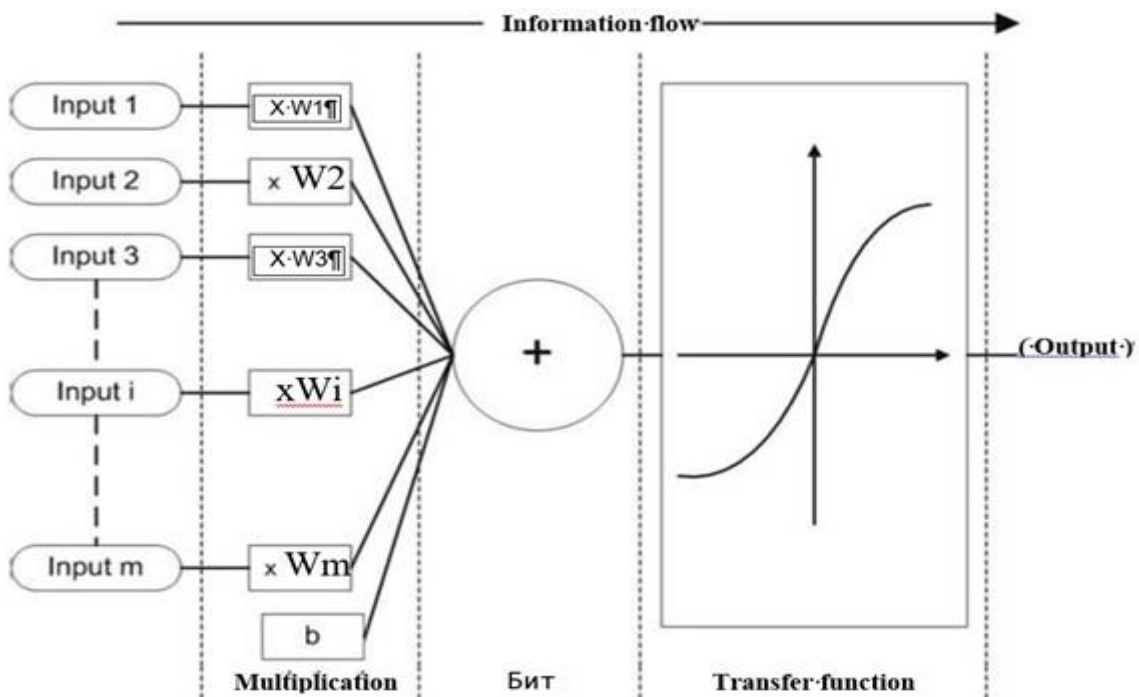


Рисунок 1.1 – Схематичне зображення штучного нейрону

Хоча правила роботи та прості набори штучних нейронів можуть здатися чимось простим, коли ми починаємо підключати нейрони до штучних нейронних мереж, відкривається повний потенціал й обчислювальна потужність таких моделей. Такі штучні нейронні мережі використовують простий факт, що складність може вирости з кількох базових правил [2] (рис. 1.2).

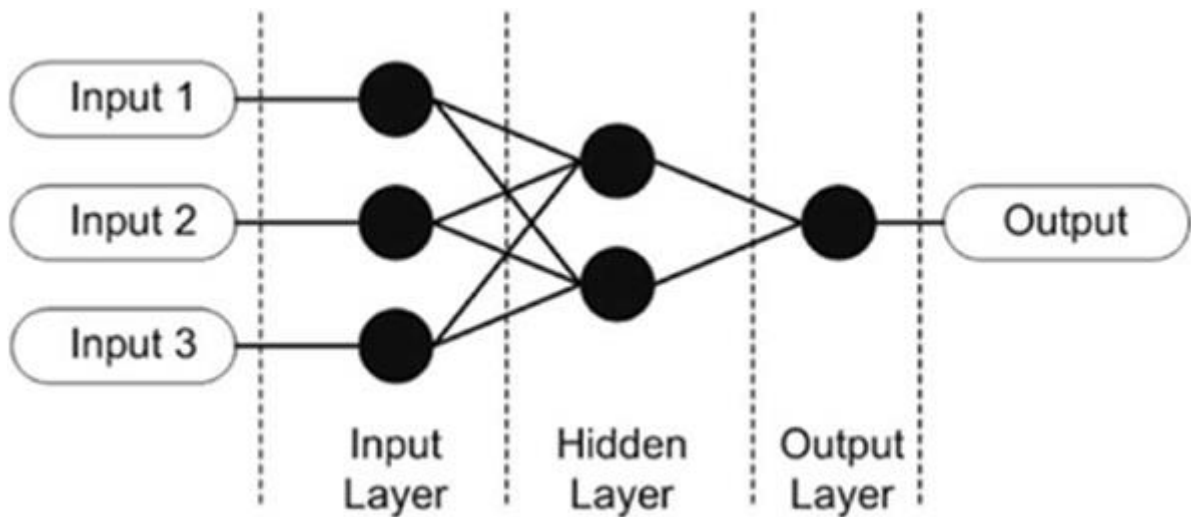


Рисунок 1.2 – Штучна нейронна мережа

Щоб повною мірою скористатися перевагами математичної складності, якої можна досягти шляхом з'єднання окремих штучних нейронів, а не просто зробити систему складною та некерованою, ми зазвичай не з'єднуємо ці штучні нейрони випадковим чином. У минулому дослідники розробили кілька стандартизованих топологій штучних нейронних мереж. Ці попередньо визначені топології допомагають вирішувати проблеми легше, швидше та ефективніше. Різні типи штучних нейронних мереж підходять для вирішення різних типів завдань. Визначивши тип поставленої задачі, необхідно визначити топологію штучної нейронної мережі, яку ми будемо використовувати, а потім налаштувати її. Нам потрібно точно налаштувати саму топологію та її параметри.

Нейронна мережа Розенблата складається з трьох елементів [3]:

- S-елемент (відчуття) – чутливий елемент, який під дією будь-якого виду енергії (світла, звуку, тиску, тепла тощо), рівень якого перевищує певний поріг T ,

створює сигнал, що дорівнює 1, інакше – 0;

- елемент (асоціація) – логічний елемент, вихідний сигнал якого дорівнює 1, коли алгебраїчна сума його вхідних сигналів дорівнює або перевищує певне граничне значення T ;

- R-елемент (відповідь) – елемент, який видає сигнал 1, якщо сума його вхідних сигналів строго позитивна, і видає сигнал 1, якщо сума його вхідних сигналів строго негативна. Якщо сума вхідних сигналів дорівнює нулю, вихід вважається нульовим або невизначеним [4].

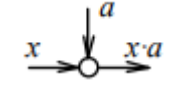

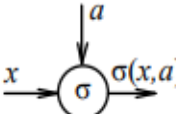

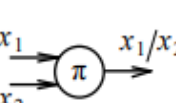

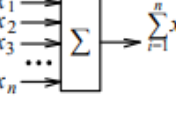
Перцептрон Розенблата є одношаровим, тому що ваги між елементами S і A фіксовані і не беруть участі в подальшому навчанні (тому до нього застосовується правило лінійного розділення).

Незважаючи на те, що ці ваги забезпечують вирішення більшості лінійно нероздільних задач [5], включаючи так звану задачу «виключного або», формулювання якої також традиційно приписується Мінському, хоча в книзі «Перцептрон» не згадується Це безпосередньо проблема.

Однією з проблем розвитку нейронних мереж є відсутність єдиної термінології та графіки, оскільки немає навіть чіткої відповіді на питання, скільки шарів має нейронна мережа, зображена на рисунку 1.2.

Крім того, як згадувалося раніше, перцептрон Розенблата не можна вважати одношаровим, хоча більшість сучасних посібників підкреслюють одношаровість при визначенні перцептронів [6] (таблиці 1.1).

Таблиця 1.1 – Основні елементи нейронних мереж

	Синапс, має два входи: вхід сигналу x та вхід синаптичної ваги a , вихідний сигнал буде рівний добутку цих значень.
	Точка розгалуження, має один вхід x та декілька виходів x_1, x_2, x_n . Причому $X_i = X$.
	Нелінійна сигмоїдальна функція активації, має один вхідний сигнал x та один параметр a .
	Пороговий перетворювач, який реалізує функцію визначення знаку.
	Нелінійний Паде перетворювач. Якщо вхідні сигнали x_1, x_2 , то вихідний сигнал буде рівний x_1/x_2
	Множувач. Має два входи та реалізує функцію множення.
	Суматор. Якщо x_1, x_2, \dots, x_n входи суматора, то на вихід рівний $\sum_{i=1}^n x_i$

1.2 Особливості створення штучного нейрону, способи навчання та оптимізації

Штучні нейрони є основними будівельними блоками кожної штучної нейронної мережі. Його дизайн і функції впливають зі спостережень за біологічними нейронами, фундаментальними будівельними блоками біологічних нейронних мереж (систем), включаючи мозок, спинний мозок і периферичні ганглії. Подібність у конструкції та функціях можна побачити на рисунку 1.3, де ліва частина малюнка представляє біологічний нейрон із сомою, дендритами та аксонами, а права частина малюнка представляє штучний нейрон із входами, вагами, функціями передачі Нейрони, зміщення та результати [7].

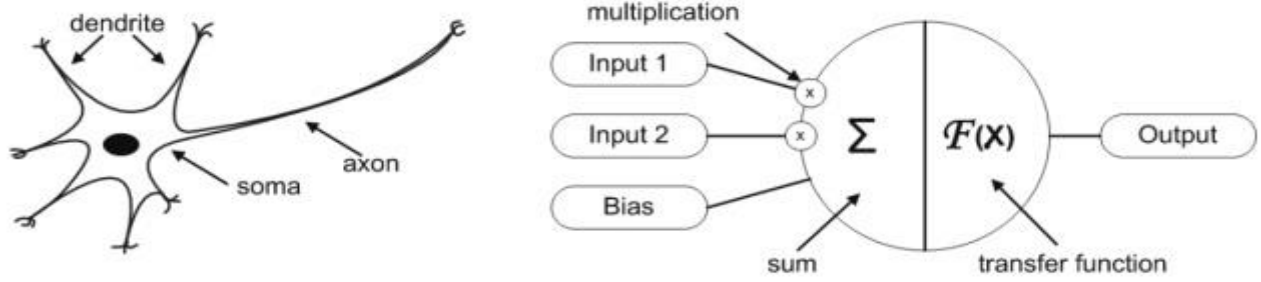


Рисунок 1.3 – Порівняння біологічного та штучного нейрону

У випадку біологічних нейронів інформація надходить до нейрона через дендрити, тіло клітини обробляє інформацію та передає її через аксон. У випадку штучних нейронів інформація надходить у тіло штучного нейрона за допомогою зважених вхідних даних [8] (кожен вхідний сигнал має бути окремо помножений на вагу). Потім тіло штучного нейрона підсумовує зважені вхідні дані, зміщення та обробляє суму за допомогою функції передачі. Нарешті, штучний нейрон передає оброблену інформацію, використовуючи вихід (i_B). Перевагу простоти моделі штучного нейрона можна побачити в її математичному описі в формулі 1.1:

$$y = g \left(\sum_{i=0}^n w_i * x_i + b \right), \quad (1.1)$$

де y – вихід,

g – функція активації,

w_i – ваги,

x_i – входи,

n – кількість з'єднань,

b – зсув.

Як видно з моделі штучного нейрона та її рівнянь, основною невідомою змінною нашої моделі є його функція активації. Функція активації визначає властивості штучного нейрона і може бути будь-якою математичною функцією. Деякі бажані властивості функцій активації включають нелінійні.

Коли функція активації є нелінійною, можна вважати, що дворівнева нейронна мережа є апроксиматором універсальної функції [9]. Функції активації ідентифікації не задовольняють цій властивості. Коли кілька рівнів використовують функцію активації ідентифікації, уся мережа еквівалентна однорівневій моделі. Формули 1.2 і 1.3 дають спрощене пояснення:

$$\begin{aligned} a^{(1)} &= x, \\ a^{(2)} &= g(W^{(1)}a^{(1)} + b^{(1)}), \\ a^{(3)} &= g(W^{(2)}a^{(2)} + b^{(2)}). \end{aligned} \quad (1.2)$$

Якщо активаційна функція δ лінійна, наприклад $\delta(x) = x$ то:

$$a^{(3)} = W^{(2)}a^{(2)} + b^{(2)} = W^{(2)}(W^{(1)}a^{(1)} + b^{(1)}) + b^{(2)} = Wx + b. \quad (1.3)$$

Труднощі, з якими стикаються системи, які покладаються на жорстко закодовані знання, дозволяють стверджувати, що системам штучного інтелекту потрібна здатність навчатися шляхом вилучення шаблонів із необроблених даних. Ця можливість називається машинним навчанням.

Організовані в групи алгоритми машинного навчання, на основі бажаного результату алгоритму [10]. Методи машинного навчання і прийняття рішень є фінальною стадією в розпізнаванні образів. Вони знаходяться на стику математичної статистики, методів оптимізації та класичних математичних дисциплін, але і має специфіку власну, пов'язану з проблемами обчислювальної ефективності та перенавчання. Загальні типи алгоритмів включають [11]:

- навчання з учителем (навчання під наглядом, supervised learning) – алгоритм генерує функцію, яка відображає вхідні дані та бажаний вихід (результат). Стандартним формулюванням навчання під контролем є класифікаційні завдання: учень повинен вивчити (приблизну поведінку) функцію, яка відображає вектор в один із кількох класів, дивлячись на кілька прикладів входу та виходу функції [12];
- навчання без вчителя (навчання без нагляду, unsupervised learning) –

моделювання набору вхідних даних при якому навчання здійснюється безпосередньо програмою без участі людини;

- навчання з підкріпленням (reinforcement learning) – алгоритми вивчають спеціальні правила, як діяти в певному середовищі. Таким чином, що кожна дія впливає безпосередньо на середовище, і це середовище забезпечує зворотний зв'язок, який спрямовано керує алгоритмом навчання.

Ці типи навчання вважаються основними видами навчання, але можна виділити ще один – напівконтрольоване навчання (semi-supervised learning). Він поєднує позначені та немарковані дані для створення відповідних ознак або класифікаторів. Цей тип можна назвати проміжним між самостійним навчанням з учителем і навчанням без учителя.

Машинне навчання — це розробка алгоритмів, які дозволяють комп'ютерам навчатися. Навчання не обов'язково передбачає свідомість, але навчання – це проблема пошуку статистичних чи інших закономірностей у даних. Таким чином, багато алгоритмів машинного навчання дуже схожі на те, як люди підходять до завдань навчання. Однак алгоритми навчання можуть дати розуміння відносної складності навчання в різних середовищах.

Навчання з учителем. Алгоритм вивчає набір навчальних даних, що керують машиною (рис. 1.4).

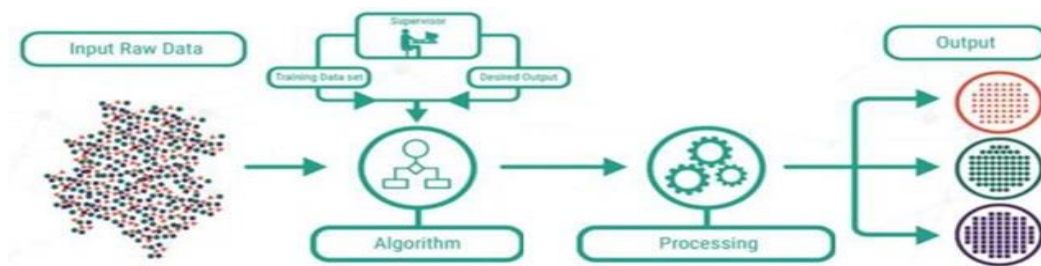


Рисунок 1.4 – Схема навчання з учителем

За допомогою контрольованого навчання вихідні дані функції передаються в систему. Це означає, що під час навчання з репетиторами машина вже знає вихід функції до того, як почне її обробляти або вивчати з неї [13]. Яскравим прикладом

цієї концепції є студент, який вивчає курс. Учень знає, що він/вона дотримується програми. З вихідним значенням функції все, що системі потрібно зробити, це визначити кроки або процеси, необхідні для досягнення того самого відображення від входу до виходу.

Якщо раптом процес не у той бік, та алгоритми виходять з результатів, які сильно відрізняються від того, що слід було очікувати, то дані тренувань спрямовують алгоритм на правильний шлях [15]. Машинне навчання з репетиторами наразі становить більшість машинного навчання, що використовується в системах по всьому світу. Вхідна змінна (x) використовується для підключення до вихідної змінної (y) за допомогою алгоритму. Вся інформація, вихідні дані, алгоритми та сценарії надаються людьми. Ми можемо краще зрозуміти контрольоване навчання, розглядаючи його через два класи проблем:

- Класифікація: розпізнавання цифр є типовим прикладом навчання класифікації. Загалом, навчання класифікації застосовне до будь-якої проблеми, де результати класифікації є корисними, а класифікацію легко визначити.

- Регресія: проблеми, які можна класифікувати як проблеми регресії, включають типи даних, де вихідні змінні встановлені на дійсні числа. Формат цієї задачі зазвичай має лінійний формат.

Алгоритми керованого навчання працюють із наборами даних, які містять функції, але кожен приклад також пов'язаний із міткою чи метою. Наприклад, набір даних «Ірис» позначено різними ірисами. Запропонований алгоритм навчання вивчає набір даних «Ірис» і навчається класифікувати рослини ірисів на три різні види на основі вимірювань.

Навчання без учителя. Хоча неконтрольоване навчання ще не впроваджено в масштабах, цей підхід формує майбутнє машинного навчання та його можливості.

Ми завжди говоримо, що машинне навчання дає безмежні можливості для майбутнього, але ми не можемо зрозуміти конкретних деталей. Щоразу, коли люди говорять про те, що комп'ютери та машини розвивають здатність ідеально «самонавчатися», вони говорять про навчання без вчителя [16].

У неконтрольованому навчанні система не має певного набору даних, і

результат більшості проблем здебільшого невідомий. Простіше кажучи, системи ШІ та об'єкти машинного навчання сліпі, коли справа доходить до виконання операцій.

Система має свої невразливі величезні логічні операції, але відсутність належних алгоритмів введення та виведення робить процес більш складним [17]. Неконтрольоване навчання здатне інтерпретувати та знаходити рішення для нескінченних масивів даних, використовуючи вхідні дані та двійкові логічні механізми, які існують у всіх комп'ютерних системах. Система взагалі не позначає дані.

Було б доречно зробити це ще більш простим, використовуючи такий приклад. Нехай є певний цифровий образ, на якому присутні певні різнокольорові геометричні фігури. Ці геометричні фігури повинні бути об'єднані в спільні групи відповідно до кольорів та інших функцій класифікації, форма тощо.

Для системи, яка працює через навчання з учителем, цей процес цілком простий. Процедура надзвичайно проста, тому що вчитель просто повинен навчити комп'ютер всім деталям, що відносяться до геометричних фігур. Система може навчитись, що всі форми з чотирма сторонами відомі як прямокутники, а інші з п'ятью сторонами називаються п'ятикутники тощо. Система також може навчитись інтерпретувати кольори і розпізнавати, як класифікується світло [18] (рис 1.5).

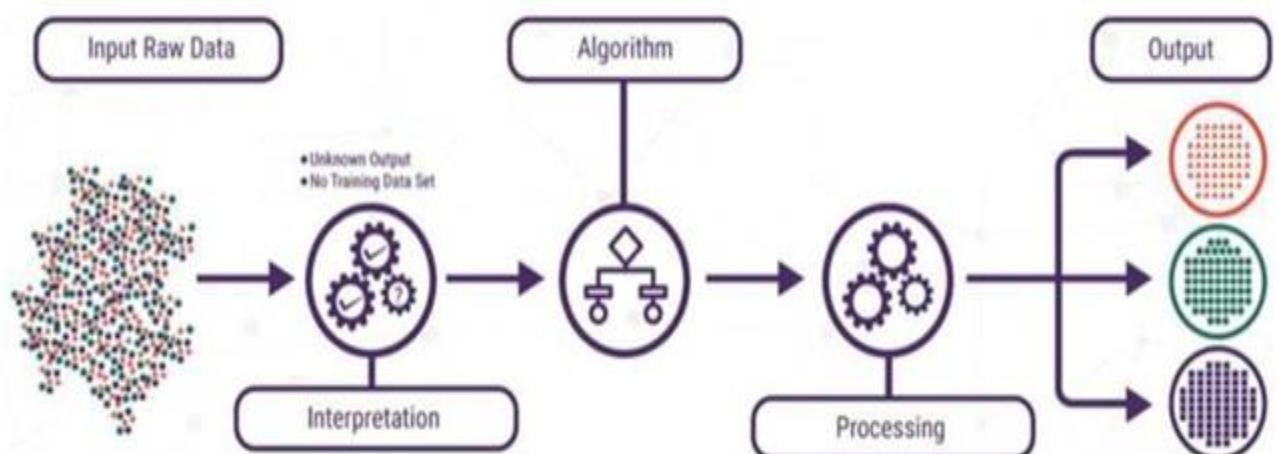


Рисунок 1.5 – Схема навчання без учителя

Проте у навчанні без вчителя весь процес стає трохи складнішим. Алгоритм процесу навчання без вчителя має ті ж вхідні дані, що і для контрольованого аналога (у нашому випадку цифрові зображення, що відображають форми в різних кольорах).

Після того, як система введе дані, вона дізнається все з наявної інформації. Фактично, сама система може визначити проблеми класифікації та відмінності у формі та кольорі. Озброївшись інформацією, що стосується проблеми, неконтрольована система може ідентифікувати всі подібні об'єкти та згрупувати їх разом. Мітки, які вони надають цим об'єктам, будуть розроблені самими машинами. Технічно можуть бути неправильні відповіді, оскільки існує певна ймовірність. Однак, як і у людей, сила машинного навчання полягає в його здатності визнавати помилки, вчитися на них і, зрештою, наступного разу приймати кращі рішення.

Алгоритм машинного навчання бере набір даних, що містить багато функцій, а потім вивчає корисні властивості структури цього набору даних. У контексті глибокого навчання ми часто хочемо вивчити весь розподіл ймовірностей, згенерований набором даних, явно в оцінці щільності або неявно для таких завдань, як синтез або зменшення шуму. Деякі алгоритми неконтрольованого навчання виконують інші ролі, такі як кластеризація, яка передбачає поділ набору даних на кластери подібних прикладів [19].

навчання з підкріпленням. Навчання з підкріпленням — це ще одна галузь машинного навчання, яка набула значної користі, допомагаючи машинам навчатися на основі їхніх процесів (рис. 1.6).



Рисунок 1.6 – Схема навчання з підкріпленням

Навчання з підкріпленням — це навчання того, що робити — як поєднувати ситуації з діями, щоб максимізувати числовий сигнал винагороди. Замість того, щоб повідомляти, які дії виконувати, алгоритм навчання повинен намагатися визначити, які дії є найбільш цінними. У найбільш цікавих і складних ситуаціях дія впливає не тільки на безпосередню винагороду, але й на наступну ситуацію, і на всі наступні винагороди. Ці дві особливості — пошук методом проб і помилок і затримка винагороди — є двома найважливішими відмінними рисами навчання з підкріпленням [20].

Навчання з підкріпленням відрізняється від навчання з репетиторами, яке зараз вивчається в більшості досліджень машинного навчання. Навчання з репетитором — це курс, заснований на наборі позначених прикладів, наданих знайомим зовнішнім викладачем. Кожен приклад є описом і специфікацією ситуації — міткою правильних дій, які система повинна виконати для цієї ситуації, що зазвичай визначає категорію, до якої належить ситуація [21].

Мета цього типу навчання полягає в тому, щоб система могла зробити висновок або узагальнити свої відповіді, щоб правильно поводитися в ситуаціях, яких не було в наборі для навчання.

Це важливий тип навчання, але недостатній для інтерактивного навчання. В інтерактивних завданнях часто неможливо отримати приклади бажаної поведінки, які є правильними і представляють усі ситуації, в яких повинен діяти агент. У неясних областях можна очікувати, що навчання буде найбільш корисним для агентів, які повинні мати можливість вчитися на власному досвіді.

Навчання з підкріпленням також відрізняється від того, що дослідники машинного навчання називають неконтрольованим навчанням, яке зазвичай передбачає пошук структури, прихованої в немаркованих наборах даних. Терміни «контрольоване навчання» та «неконтрольоване навчання», здається, всебічно класифікують парадигми машинного навчання, але цього недостатньо.

Хоча спокусливо вважати навчання з підкріпленням неконтрольованим, оскільки воно не спирається на приклади правильної поведінки, навчання може

посилити сигнал винагороди, а не намагатися знайти приховану структуру. Виявлення структури в досвіді агента, безумовно, корисно для навчання з підкріпленням, але саме по собі воно не вирішує проблеми навчання з підкріпленням максимізації сигналу винагороди. Таким чином, навчання з підкріпленням можна вважати третьою парадигмою освітнього процесу поряд з навчанням з учителем і без нього та, можливо, іншими парадигмами [22].

Однією з проблем навчання з підкріпленням порівняно з іншими типами навчання є компроміс між дослідженням і експлуатацією. Щоб отримати суттєву винагороду, агент навчання з підкріпленням повинен віддавати перевагу поведінці, яку він пробував у минулому та визнав її ефективною для отримання винагороди. Але щоб виявити таку поведінку, він повинен спробувати поведінку, яку раніше не вибирав. Агент повинен використовувати досвід, який він уже має, щоб отримати винагороду [23], але він також повинен навчитися виконувати кращі дії в майбутньому.

Дилема полягає в тому, що неможливо спеціалізуватися на дослідженні чи розробці, не вирішивши проблеми. Агент повинен спробувати різні дії і поступово вибрати найкращу для себе. Відповідно до випадкового завдання, кожен дію слід повторювати багато разів, щоб отримати надійну оцінку очікуваної винагороди. Дилема «розвідка-експлуатація» інтенсивно вивчається математиками протягом десятиліть, але залишається невирішеною. Наразі ми просто зазначаємо, що вся проблема балансу дослідження та експлуатації навіть не виникає в навчанні під наглядом і без репетитора, принаймні в його найпоширенішій формі.

Іншою ключовою особливістю навчання з підкріпленням є те, що воно чітко вирішує всю проблему взаємодії цілеспрямованих агентів із невизначеним середовищем. Це різко контрастує з багатьма підходами, які зосереджуються лише на підзавданнях без урахування того, як вони вписуються в загальну картину.

Наприклад, було згадано, що багато досліджень у галузі машинного навчання пов'язані з навчанням з учителем, але чітко не визначено, як ця здатність зрештою буде корисною. Інші дослідники розробили теорії планування із загальними цілями, але без розгляду ролі планування в процесах прийняття рішень у реальному часі або

питання про те, де будуть зроблені прогностичні моделі, необхідні для планування. Хоча ці методи дають багато корисних результатів, їхня зосередженість на проблемі ізольованих підпроблемах є суттєвим обмеженням [24].

Навпаки, навчання з підкріпленням починається з повного інтерактивного агента, який шукає ціль. Усі агенти навчання з підкріпленням мають чіткі цілі, можуть сприймати аспекти середовища та можуть вибирати дії, які впливають на середовище.

Крім того, з самого початку часто припускають, що агент повинен виконати незважаючи на значну невизначеність середовища, з яким він стикається. Коли навчання з підкріпленням передбачає планування, воно має враховувати взаємодію між плануванням і вибором дій у реальному часі, а також як покращити моделі середовища. Коли навчання з підкріпленням передбачає навчання з учителем, конкретні причини визначають, які навички є критичними, а які ні.

Для прогресу дослідження важливі підзадачі повинні бути виділені та вивчені, але вони повинні бути підзавданнями, які відіграють чітку роль у повному, інтерактивному агенті пошуку мети, навіть якщо всі деталі повного агента ще не можуть бути завершені [25] (рис. 1.7).

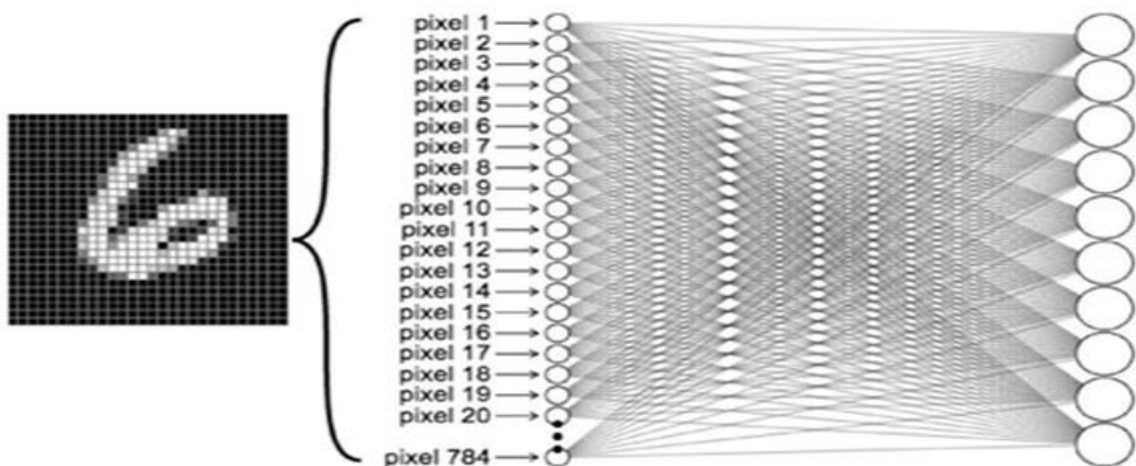


Рисунок 1.7 – Одношарова нейронна мережа для розпізнавання рукописних цифр

Під повним інтерактивним агентом пошуку цілі ми не завжди маємо на увазі

щось схоже на повний організм або робота. Це очевидний приклад, але повний, інтерактивний агент пошуку мети також може бути невід’ємною частиною більшої поведінкової системи. У цьому випадку агент взаємодіє безпосередньо з іншою більшою системою і опосередковано з середовищем більшої системи. Простим прикладом є агент, який відстежує рівень заряду батареї робота та надсилає команди до архітектури керування роботом. Середовищем агента є робот, а також середовище робота. Необхідно вийти за рамки найбільш очевидного прикладу агента та його середовища, щоб зрозуміти загальну структуру навчання з підкріпленням.

Важливою складовою нейронної мережі є візуалізація. Щоб проаналізувати цей етап роботи нейронної мережі візьмемо мережу, для того, щоб класифікувати рукописні цифри ММБТ, та створимо сполучення безпосередньо від вхідного шару до вихідного шару без прихованих шарів між ними [26]. Таким чином, наша мережа виглядає так, як на рисунку 1.7, наведеному вище.

Згадаємо, коли вводять зображення в нашу нейронну мережу, ми візуалізуємо мережеву діаграму, «розгорнувши» пікселі в один стовпець нейронів, як показано на рисунку 1.8. Давайте зосередимо увагу лише на з’єднанні, підключеному до першого вихідного нейрона, який ми будемо позначати Z , і позначатимемо кожен з вхідних нейронів та їхні відповідні ваги як x_i та W_i .

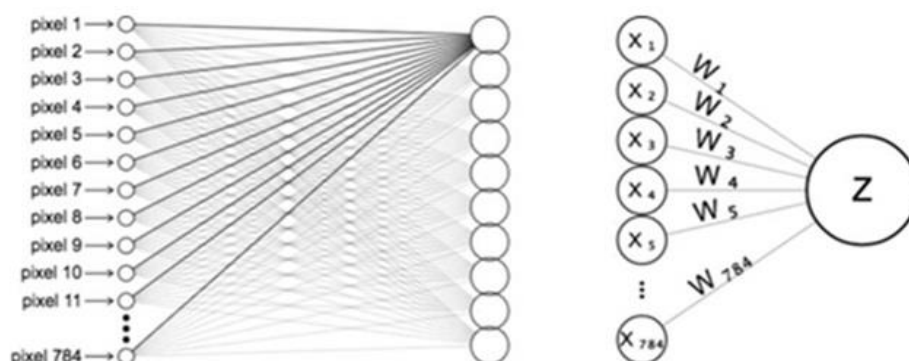


Рисунок 1.8 – З’єднання до вихідного нейрона

Замість того, щоб розгортати пікселі, краще розглянемо сітку вагів розміром 28 x 28, де ваги розташовані точно так само, як і відповідні пікселі. Представлення наведено на попередньому рисунку 1.8 виглядає інакше, ніж на рисунку 1.9, але вони виражають те саме рівняння.

$$\begin{array}{cccc}
 X_1 & X_2 & X_3 & \dots & X_{28} \\
 X_{29} & X_{30} & X_{31} & & X_{56} \\
 X_{57} & X_{58} & X_{59} & & X_{84} \\
 \vdots & & & \ddots & \\
 X_{757} & X_{758} & X_{759} & & X_{784}
 \end{array}
 \circ
 \begin{array}{cccc}
 W_1 & W_2 & W_3 & \dots & W_{28} \\
 W_{29} & W_{30} & W_{31} & & W_{56} \\
 W_{57} & W_{58} & W_{59} & & W_{84} \\
 \vdots & & & \ddots & \\
 W_{757} & W_{758} & W_{759} & & W_{784}
 \end{array}
 =
 \begin{array}{c}
 \bigcirc \\
 Z
 \end{array}$$

Рисунок 1.9 – Візуалізація множення пікселів на ваги

Давайте тепер візьмемо навчену нейронну мережу з цією архітектурою та візуалізуємо навчені ваги першого вихідного нейрона, який відповідає за класифікацію нульової цифри [27]. Позначимо їх кольором, тобто найменше значення ваги буде чорним, а найвище значення – білим (рис. 1.10).

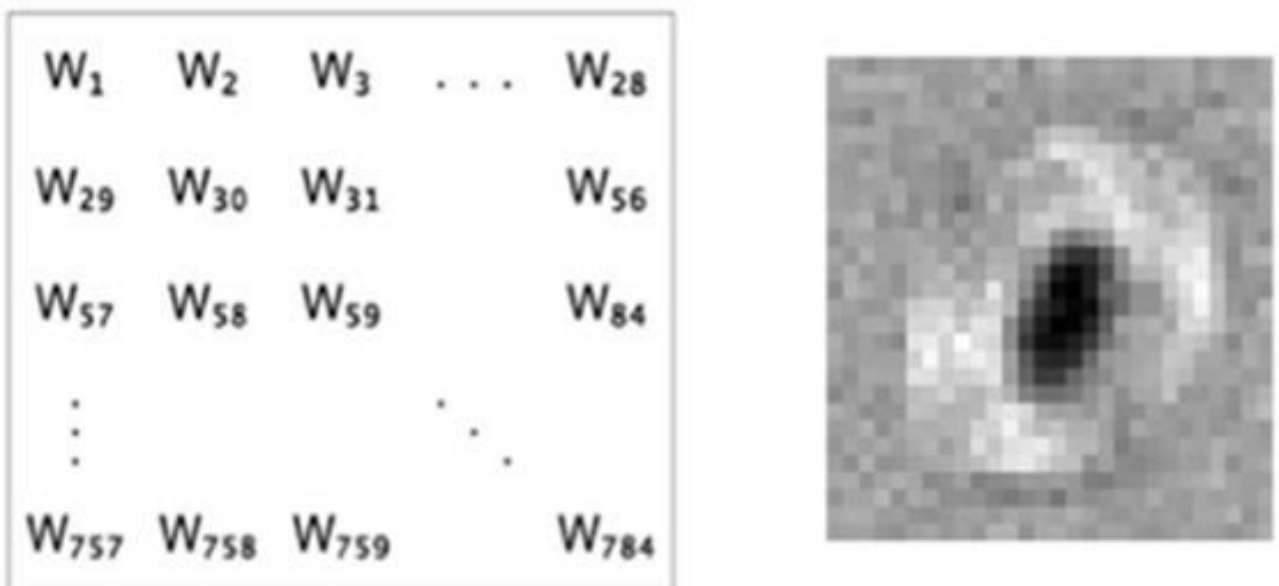


Рисунок 1.10 – Візуалізація ваг вихідного нейрона, який відповідає за цифри 0

Нуль виглядає трохи нечітким. Чому це так, стане зрозуміліше, якщо ми розглянемо, що роблять нейрони. Оскільки він відповідає за класифікацію нулів, його метою є виведення високих значень для нулів і низьких значень для ненульових.

Він може досягти високого нульового результату, вирівнюючи великі ваги з пікселями в нульовому зображенні, які, як правило, мають високі значення. У той же час він може досягти відносно низького ненульового результату, маючи малі вагові коефіцієнти вирівнювання пікселів, які мають тенденцію бути вищими на ненульових зображеннях і нижчими на нульових зображеннях. Зображення відносно чорного центру шкали залежить від того факту, що знімки нуля мають тенденцію відхилятися тут (дірки в межах нуля), але інші числа зазвичай вищі [28]. Давайте подивимося на ваги, отримані для всіх 10 вихідних нейронів (рис. 1.11).

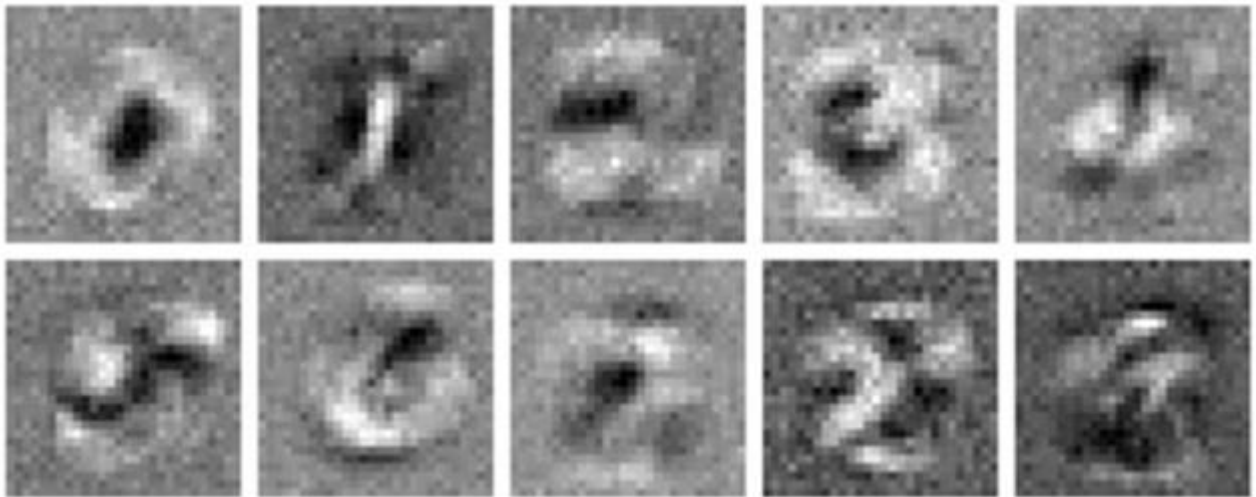


Рисунок 1.11 – Візуалізація ваг вихідних нейронів

Як очікувалося, всі вони виглядають дещо розмитими версіями наших десяти цифр. Вони з'являються майже так, якби ми усереднили багато зображень, що належать до кожного класу цифр.

Припустимо, ми отримуємо вхідні дані з другим зображенням. Ми можемо

передбачити, що нейрон, відповідальний за класифікацію двійкових файлів, повинен мати високе значення, оскільки він має таку вагу, що високі ваги мають тенденцію збігатися з пікселями, які мають тенденцію бути високими у двійкових файлах. Для інших нейронів деякі вагові коефіцієнти також відповідатимуть високим значенням пікселів, які мають трохи вищий бал. Однак перекриття буде набагато менше, і багато з цих зображень з високими значеннями пікселів будуть зміщені через низькі ваги в інших нейронах. Функція активації не змінює цього, оскільки вона монотонна відносно входу, тобто чим вищий вхід, тим вищий вихід [29].

Ми можемо інтерпретувати ці ваги як шаблони, які формують вихідні категорії. Це дійсно цікаво, тому що ми ніколи не повідомляємо мережі заздалегідь, що це за числа або що вони означають, але вони все одно нагадують об'єкти цих класів. Це натякає на щось особливо важливе всередині нейронних мереж: вони формують представлення об'єктів, які вони навчають, і виявляється, що ці представлення можуть бути використані для набагато більшого, ніж проста класифікація чи передбачення [30].

Це викликає набагато більше питань, ніж це дає відповіді, такі як, що відбувається з вагами при додаванні прихованих шарів? Як приховані шари впливають на нашу нейронну мережу? Щоб побачити, спробуємо вставити прихований шар десяти нейронів у нашу мережу. Отже, наша нейронна мережа для класифікації рукописних цифр виглядає як показано на рисунку 1.12.

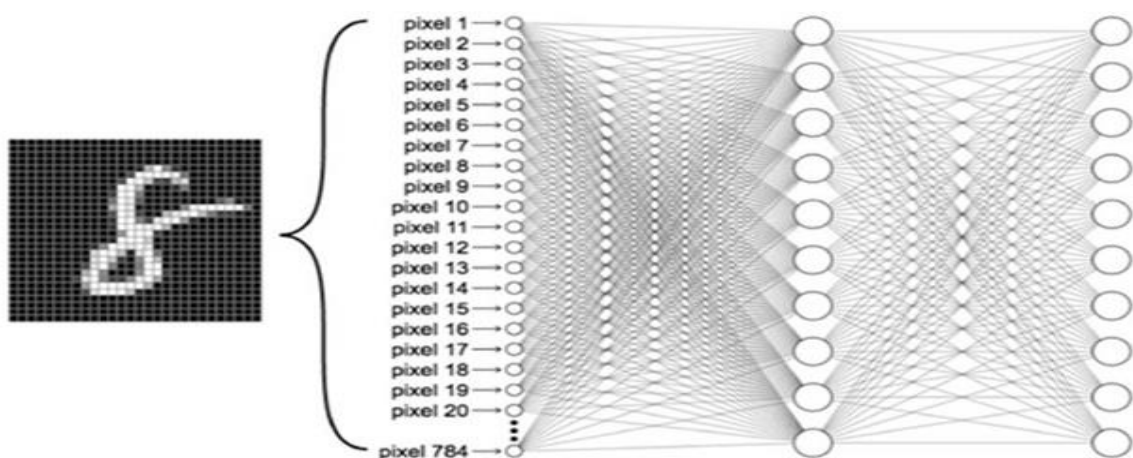


Рисунок 1.12 – Двошарова нейронна мережа для розпізнавання рукописних цифр

Наша проста метафора шаблону однорівневої мережі, наведена вище, не застосовується в цьому випадку, оскільки ми більше не маємо 784 вхідних пікселів, безпосередньо підключених до вихідного класу. У певному сенсі ми можемо сказати, що ми «примушуємо» нашу оригінальну одношарову мережу вивчати ці шаблони, оскільки кожна вага безпосередньо пов'язана з міткою класу i , таким чином, впливає лише на цей клас. Але в складній мережі, яку ми зараз представляємо, ваги прихованого шару впливають на всі десять нейронів вихідного шару. Отже, як ви можете очікувати, що ці ваги виглядатимуть зараз.

Щоб зрозуміти, що відбувається, ми візуалізуємо вагові коефіцієнти на першому шарі, як і раніше, але ми також уважніше розглянемо, як їхні активації поєднуються на другому шарі, щоб отримати оцінку класу. Нагадаємо, що якщо зображення значною мірою «симпатизує» цьому фільтру [31], зображення вироблятиме високу активацію в певному нейроні першого шару.

Таким чином, десять нейронів у прихованому шарі представляють присутність цих десяти особливостей у вихідному зображенні. На вихідному рівні нейрон, що відповідає значенню класу, є зваженою комбінацією перших десяти прихованих активацій. Візуалізація ваг прихованих і вихідних шарів і активації нейронів показана на рисунку 1.13.

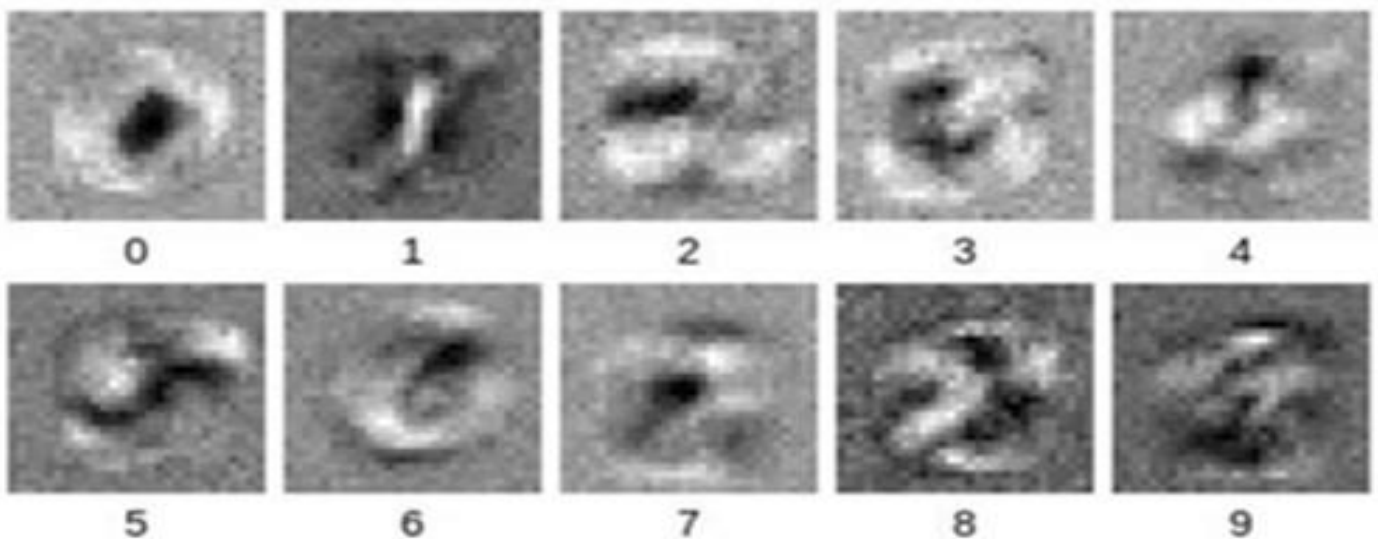


Рисунок 1.13 – Візуалізація ваг першого (прихованого) шару нейронів

Почнемо з масштабу першого шару, візуалізованого у верхній половині. Вони більше не схожі на шаблони, схожі на зображення, але й не знайомі. Одні з них виглядають як псевдоцикла, а інші — як будівельні блоки цих чисел: півкільця, діагоналі, отвори тощо.

Рядки під відфільтрованими зображеннями відповідають нашим вихідним нейронам, по одному для кожної категорії зображень.

Ці стовпці представляють ваги, пов'язані з кожним із десяти активацій фільтра з прихованого шару. Наприклад, клас «0» підтримує фільтри вищого рівня вздовж зовнішнього контуру (де число нуль має тенденцію з'являтися).

Це пригнічує деякі фільтри з високими пікселями посередині (що зазвичай є отвором). Рівень «1» є майже протилежним цьому значенню, віддаючи перевагу сильнішому фільтру посередині, де можна побачити рівномірні вертикальні смуги [32].

Перевагою цього підходу є гнучкість. Для кожного класу існує ширший діапазон вхідних шаблонів, які стимулюють відповідні вихідні нейрони. Кожен клас можна викликати за допомогою кількох абстрактних функцій, наявних у попередньому прихованому шарі, або їх комбінації. По суті, ми можемо вивчати різні типи нулів, різні типи тощо для кожного класу. Зазвичай (але не завжди) це покращує продуктивність мережі для більшості завдань (рис. 14).

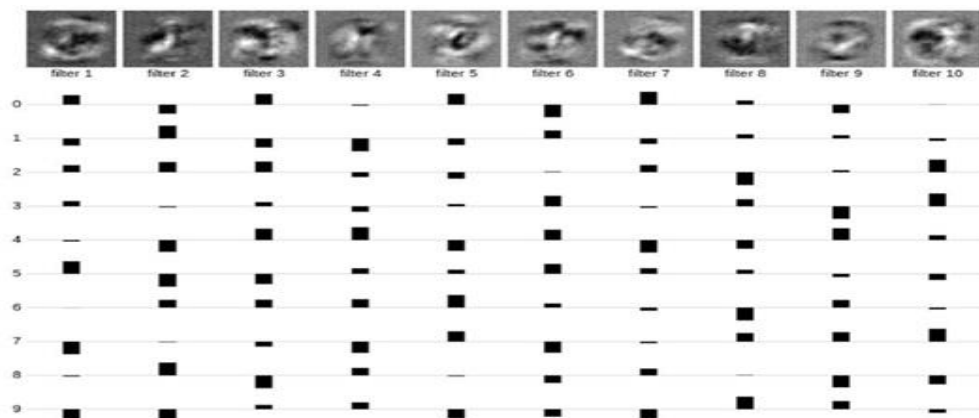


Рисунок 1.14 – Візуалізація ваг другого шару нейронів та активації для кожного

класу

В одношарових і багатошарових нейронних мережах кожен рівень виконує однакову функцію; він перетворює дані з попереднього рівня в представлення цих даних "вищого рівня". Під «вищим рівнем» я маю на увазі, що він містить компактне та більш виразне представлення цих даних, оскільки зведення є представленням «високого рівня». Наприклад, у наведеній вище 2-рівневій мережі ми зіставляємо пікселі низького рівня з високорівневими функціями на першому шарі графіки (штрихи, петлі тощо), а потім зіставляємо ці високорівневі функції з ієрархічною структурою вищого рівня. представлення в наступному шарі. Ця концепція перетворення даних у меншу, але більш значущу інформацію є основою машинного навчання та основною можливістю нейронних мереж [33].

Додаючи прихований рівень до нейронної мережі, ми даємо їй можливість вивчати функції на різних рівнях абстракції. Це дає нам насичене представлення даних, де ми маємо низькорівневі функції на ранніх рівнях і високорівневі функції на наступних рівнях, що складаються з ранніх функцій шару.

Як ми бачимо, приховані шари можуть підвищити точність, але лише в обмеженій мірі. Додавання все нових і нових шарів швидко перестає підвищувати точність і вимагає обчислювальних витрат — ми не можемо просто попросити нашу нейронну мережу запам'ятати всі можливі версії класу зображення через його приховані шари [34].

1.3 Огляд нейронних мереж для створення зображень

Нейронна мережа прямого поширення (feed forward neural networks, FFNN) і персептони передають інформацію з передньої до задньої (вхід і вихід, відповідно). Нейронні мережі часто описуються такі, що мають шари, де кожен окремий шар складається з паралельних вхідних, прихованих або вихідних комірок. Шар сам по собі не має жодних зв'язків і загалом два сусідні шари повністю пов'язані (кожен нейрон утворює один шар для кожного нейрона до іншого шару) [35]. Найпростіша

дещо практична мережа має два вхідні нейрони та один вихідний, які можна використовувати для моделювання логічних воріт.

Один, як правило, готує FFNN за допомогою зворотного поширення надайте мережевий набір даних про те, «що сталося» і «що ми хочемо отримати». Це називається контрольованим навчанням, на відміну від неконтрольованого навчання, коли ми просто вводимо дані та дозволяємо мережі заповнити прогалини. Помилка зворотного поширення зазвичай є певною зміною різниці між вхідним і вихідним сигналами (наприклад, середньоквадратична помилка або проста лінійна різниця). Враховуючи, що мережа має достатньо прихованих нейронів, вона теоретично завжди може моделювати зв'язок між входом і виходом. На практиці їх використання дуже обмежене, але вони широко комбінуються з іншими мережами для формування нових мереж [36].

Мережа Хопфілда (Hopfield network, HN) – це мережа, в якій кожен нейрон з'єднаний з кожним іншим нейроном; це абсолютно заплутане як "блюдо зі спагетті", тому що навіть усі вузли можуть бути всім. Кожен окремий вузол вводиться перед навчанням, потім ховається під час навчання та виводиться після навчання. Мережа навчається шляхом встановлення значень нейронів у потрібний режим, після чого можна обчислити ваги. Ваги після цього не змінюються рис (1.15).



Рисунок 1.15 – Структура нейронної мережі прямого поширення

Після навчання на одній або кількох моделях мережа завжди відповідатиме одній із вивчених моделей, оскільки мережа стабільна лише в цих станах. Зауважте, що це не завжди бажаний стан.

Кожен нейрон має поріг активації, що дорівнює цій температурі, яка, якщо сума вхідних даних перевищує цей поріг, змушує нейрон приймати один із двох станів (зазвичай 1 або 1, іноді 0 або 1).

Оновлення мережі може виконуватися синхронно або частіше один за одним. Якщо оновлено один за одним, то створюється чесна випадкова послідовність для того, щоб організувати, які ярлики оновлювати в якому порядку (справедливий випадковий всі параметри (n), що відбуваються рівно раз на кожні n елементів) [37] (рис. 16).

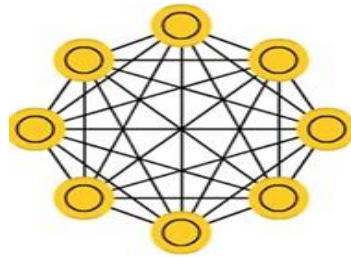


Рисунок 1.16 – Структура мережі Хопфілда

Таким чином, ви можете визначити, коли мережа стабільна (виконується конвергенція), коли кожен нейрон був оновлений, і жодний з них не був змінений, мережа стабільна (annealed). Ці мережі часто називають асоціативною пам'яттю, тому що вони сходяться до найбільш подібного стану як вхідні; якщо можна побачити одну половину столу, ми можемо уявити другу половину.

Машини Больцмана (Boltzman Machine, BM) дуже схожі на HN, але: деякі нейрони відмічені як вхідні нейрони, а інші залишаються «прихованими» (рис. 1.17). Вхідні нейрони стають вихідними нейронами в кінці повного оновлення мережі.

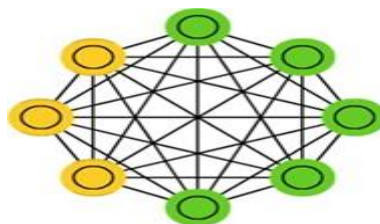


Рисунок 1.17 – Структура машини Больцмана

Воно починається з довільних ваг і навчається через так зване зворотне поширення, або останнім часом через контрастну дивергенцію (Маркові використовується для визначення градієнтів між двома інформаційними досягненнями) [38].

У порівнянні з мережею Хопфілда, нейрони більшою мірою мають двійкові візуальні активації. Машини Больцмана є стохастичними мережами. Процес тренувань та виконання машиною Больцмана досить схожий на мережу Хопфілда: він встановлює входні нейрони на певні закріплені значення, після чого мережа встановлюється довільно.

Ми виконуємо ітерації між входними та прихованими нейронами, хоча вільні клітини можуть приймати будь-яке значення. Активація контролюється глобальним значенням температури, яка зменшує енергію клітини при зниженні температури. Ця нижча енергія призводить до стабілізації їх схем активації [39].

Мережа досягає рівноваги з урахуванням потрібної температури. Автокодері (Autoencoders, AE) дещо схожі на FFNNs, оскільки AE більше схожий на модифіковану FFNN, ніж принципово іншу архітектуру. Основна ідея автокодерів полягає в тому, щоб автоматично кодувати інформацію (наприклад, стиснення, а не шифрування), звідси і назва. Вся мережа завжди схожа на пісочний годинник, прихований шар менший за форму входного та вихідного шарів. AE завжди симетричні навколо середнього шару (один або два залежно від парної чи непарної кількості шарів) [40] (рис. 1.18).



Рисунок 1.18 – Структура автоенкодера

Найменший шар(и) майже завжди знаходиться в середині, місце, де інформація найбільш стиснута. Все до середини називається частиною кодування, все після середини декодування а середина – кодом.

Їх можна навчити за допомогою зворотного поширення вхідного сигналу, причому помилка встановлюється як різниця між вхідним і вихідним сигналами. Що стосується ваг, АЕ також можна побудувати симетрично, щоб ваги кодування були такими ж, як і ваги декодування [41].

Згорткові нейронні мережі (CNN) дуже відрізняються від більшості інших мереж. Вони в основному використовуються для обробки зображень, але також можуть використовуватися для інших типів введення, наприклад аудіо. Типовим випадком використання CNN є передача зображень у мережу, а потім мережа класифікує дані: якщо вхідні дані є зображенням kota, вона виводить «кішку», а якщо вхідні дані є зображенням собаки, вона виводить «собака». Як правило, CNN мають на вході «сканер», який не призначений для аналізу всіх навчальних даних одночасно [42].

Наприклад, щоб ввести зображення розміром 200 x 200 пікселів, вам не потрібно створювати шар із 40 000 вузлів. Найкраще створити вхідний шар розміром 20 x 20, який приймає зображення розміром 20 x 20 пікселів (зазвичай починаючи з верхнього лівого кута). Після того, як дані пройшли через вхід (і, імовірно, ці дані були використані для навчання), вхідний рівень займає наступні 20 x 20 пікселів: сканер зсувається на один піксель вправо (див. також рис. 1.20).

При цьому зображення не розбивається на блоки 20 x 20, а сканер ковзає по цьому зображенню із заданим розміром кроку. Потім ці вхідні дані подаються на згорткові рівні замість звичайних рівнів, де не всі вузли підключені до всіх вузлів. Кожен вузол підключається лише до найближчих вузлів (наскільки близько залежить від реалізації, але зазвичай не більше ніж кілька вузлів). Ці згорткові шари зменшуються з глибиною (тому 20 може перейти в шар 10, а потім у шар 5). Тут часто використовуються ступені числа 2: 32, 16, 8, 4, 2, 1. На додаток до цих

згорткових шарів, згорткові нейронні мережі часто мають шари об'єднання (накопичення).

Об'єднання — це спосіб відфільтрувати дрібні деталі: загальноприйнятою технікою об'єднання є максимальне об'єднання, де ми беремо 2 x 2 пікселі та передаємо ті, у яких найбільше червоних чисел. Щоб застосувати CNN до аудіо, вхідні звукові хвилі подаються сегмент за сегментом. У реальних реалізаціях CNN зазвичай додаються до кінця FFNN для подальшої обробки даних. Ці мережі називаються DCNN, але назви цих мереж часто використовуються як синоніми [43].

Існують також такі методи класифікації типів штучних нейронних мереж.

Повністю зв'язана нейронна мережа (FCNN) повністю складається з повністю зв'язаних рівнів, які з'єднують кожен нейрон одного рівня з кожним нейроном іншого рівня. Кожен нейрон передає вихідний сигнал решті нейронів, включаючи себе. Вихідний сигнал мережі може бути вихідним сигналом усіх нейронів. Елементи ієрархічної та повнозв'язної мереж можна вибирати різними. Існує, втім, стандартний вибір: нейрон з адаптивним неоднорідним лінійним суматором на вході, схема повнозв'язної нейронної мережі продемонстрована на рисунку 1.19.

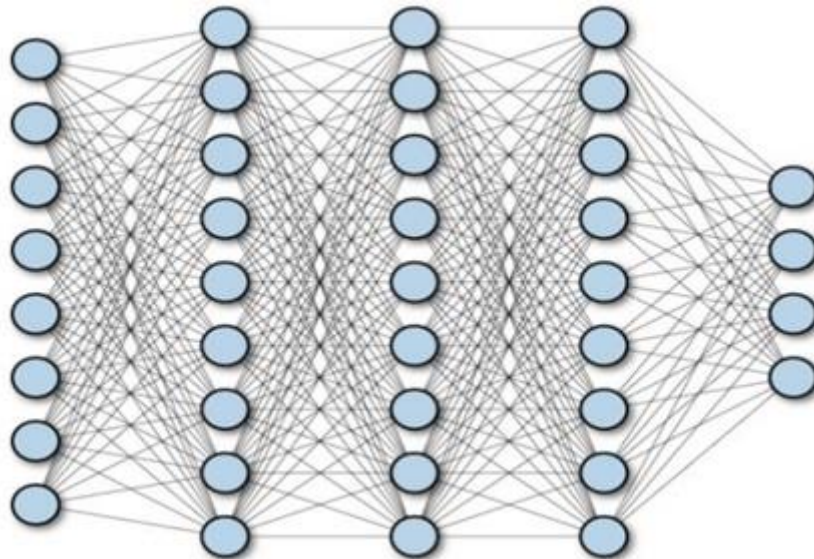


Рисунок 1.19 – Схема повнозв'язної нейронної мережі

Основна перевага повністю зв'язаних мереж полягає в тому, що вони

«незалежні від структури», тобто не роблять особливих припущень щодо вхідних даних. Для повністю зв'язаної мережі вхідний суматор нейрона фактично розділений на два: перший обчислює лінійну функцію на основі вхідного сигналу мережі, а другий обчислює лінійну функцію на основі вихідних сигналів інших нейронів, отриманий на попередньому кроці [44]. Функція активації нейрона (власна функція) є нелінійним перетворювачем вихідного сигналу суматора. Мережа називається однорідною, якщо всі її нейрони функціонують однаково. Мережа називається неоднорідною, якщо функція ознаки залежить від одного або кількох параметрів, значення яких різняться між нейронами [45].

Нейронні мережі можуть мати декілька прихованих шарів. Немає необхідності створювати стандартну мережу нейронів. Ієрархічні або повністю пов'язані архітектури не накладають істотних обмежень на елементи, які в них беруть участь. Єдиною суворою вимогою архітектури до елемента мережі є відповідне співвідношення між розмірністю вектора вхідного сигналу елемента мережі (визначається архітектурою) і кількістю входів. Незважаючи на те, що агностична структура робить повністю підключені мережі дуже широко застосовними, такі мережі, як правило, мають слабку ефективність, ніж мережі спеціального призначення, налаштовані на структуру проблемного простору [46] (рис 1.20).

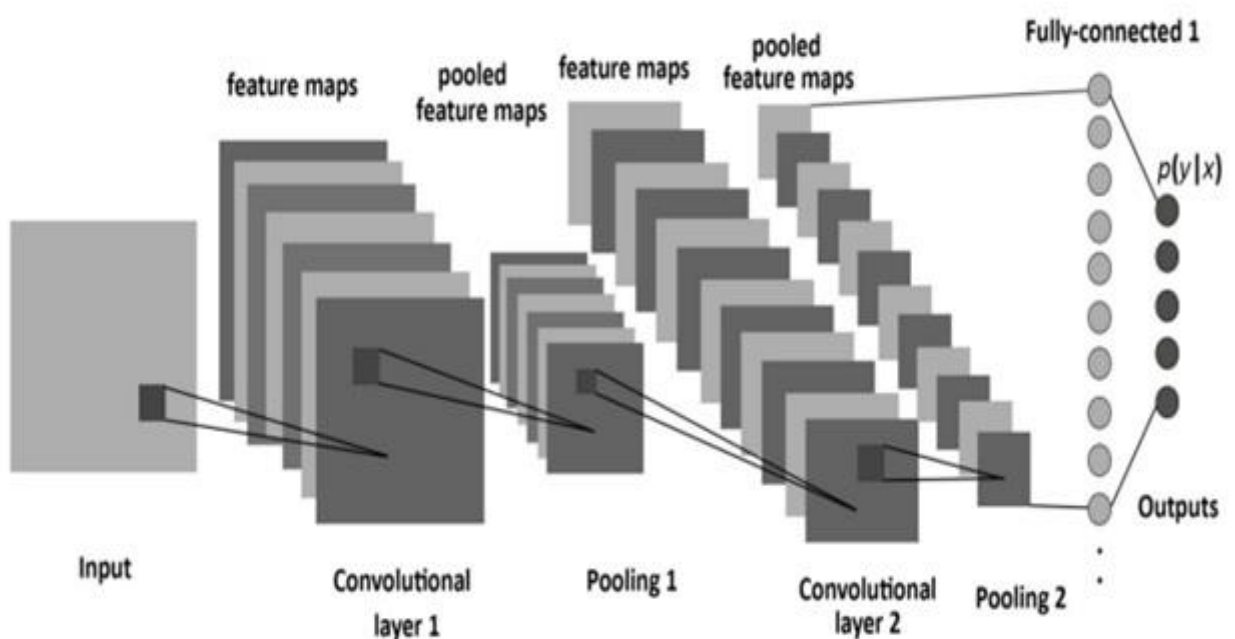


Рисунок 1.20 – Схема згорткової нейромережі

Згадувана вище згорткова нейронна мережа (Convolutional neural network, CNN) має спеціальну архітектуру, яка дозволяє їй максимально ефективно розпізнавати образи. Ідея CNN чітко висловлюють припущення, що вхідні дані – це зображення, що дозволяє кодувати певні властивості в архітектурі моделі.

Простий CNN – це послідовність шарів, і кожен шар CNN перетворює один обсяг активацій в інший через диференційовану функцію. Для побудови архітектури CNN використовуються три основні типи шарів: згортковий шар, шар об'єднання та повністю з'єднаний шар. Модель CNN ґрунтується на чергуванні згорткових і субдискретизованих шарів (pooling), а структура є односпрямованою. CNN отримала свою назву від операції згортання, яка припускає, що кожен сегмент зображення буде поелементно помножено на ядро згортки, а отримані результати повинні бути додані та записані в те саме місце, що й вихідне зображення. Ця архітектура забезпечує незмінність розпізнавання щодо зміщення об'єкта, поступово збільшуючи «вікно», в якому «бачать» звивини, виявляючи все більші й більші структури та візерунки в зображеннях [47].

Рекурсивні нейронні мережі (Recursive neural network, RvNN) – вид нейронних мереж, що працюють з даними змінної довжини. Моделі рекурсивних мереж використовують ієрархічні структури зразків при навчанні. Наприклад, зображення, складені з сцен, які об'єднують підсцени, що включають багато об'єктів. Виявлення структури сцени і її деконструкція – нетривіальне завдання. При цьому необхідно як ідентифікувати окремі об'єкти, так і всю структуру сцени (рис. 1.21).

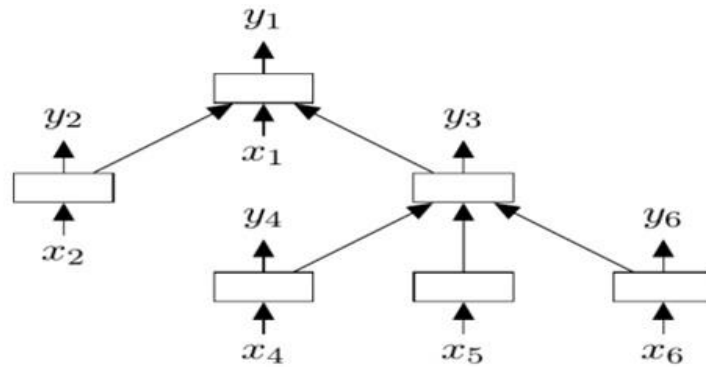


Рисунок 1.21 – Схема рекурсивної нейронної мережі

У рекурсивних мережах нейрони з однаковими вагами активуються рекурсивно відповідно до структури мережі. В процесі роботи рекурсивної мережі виробляється модель для передбачення для структур змінної розмірності, так і скалярних структур через активацію структури відповідно до топології.

Мережі RvNNs успішно застосовуються при навчанні послідовних структур і дерев в задачах обробки природної мови, при цьому фрази і пропозиції моделюються через векторне подання слів. RvNNs спочатку з'явилися для розподіленого представлення структур, використовуючи предикати математичної логіки [48].

Мережа довгої короткостроковій пам'яті – вид рекурсивної НС, що дозволяє максимально точно моделювати тимчасові послідовності, а також характерні для них залежно в довгостроковій перспективі.

Генеративно-змагальна мережа (Generative Adversarial Networks, GAN) – це алгоритм машинного навчання без учителя, побудований на комбінації з двох нейронних мереж, одна з яких (мережа G) генерує зразки (Генеративна модель), а інша (мережа D) намагається відрізнити правильні («справжні») зразки від неправильних [49] (рис. 1.22).

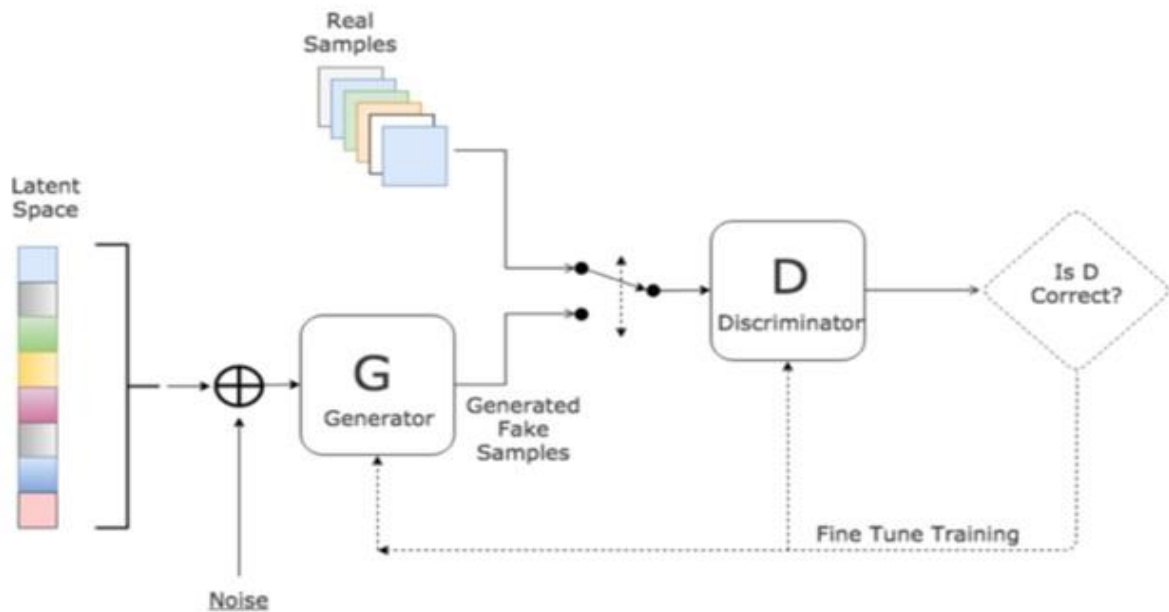


Рисунок 1.22 – Схема генеративно-змагальної мережі

Базовий GAN складається з наступних елементів:

- генеративна модель (тобто генератор) генерує об'єкт. Генератор нічого не знає про реальних об'єктах і вчиться, взаємодіючи з дискримінатором. Наприклад, генератор може генерувати зображення.

- Дискримінаційна модель (тобто Дискримінатор) визначає, чи є об'єкт реальним (зазвичай представлений значенням, близьким до 1) або підробленим (представлений значенням, близьким до 0). Дискримінатор подає генератору сигнал змагальної втрати (або помилки) таким чином, що він дозволяє генератору генерувати об'єкти, максимально наближені до реальних об'єктів [50].

РОЗДІЛ 2 ОПИС ТА АРХІТЕКТУРА МЕРЕЖІ NERF ДЛЯ СТВОРЕННЯ 3D ЗОБРАЖЕНЬ

2.1 Архітектура штучних нейронних мереж

ШНМ можна розглядати як орієнтований граф зі зваженими зв'язками, де вузлами є штучні нейрони. Залежно від архітектури з'єднань, ШНС можна розділити на дві категорії: мережі прямого поширення, де граф не має петель, і рекурентні мережі, або мережі зі зворотним зв'язком.

У найбільш поширеному сімействі мереж першого типу, які називаються багат шаровими перцептронами, нейрони розташовані шарами з одностороннім зв'язком між шарами. На рисунку 2.1 представлені типові мережі кожного класу.

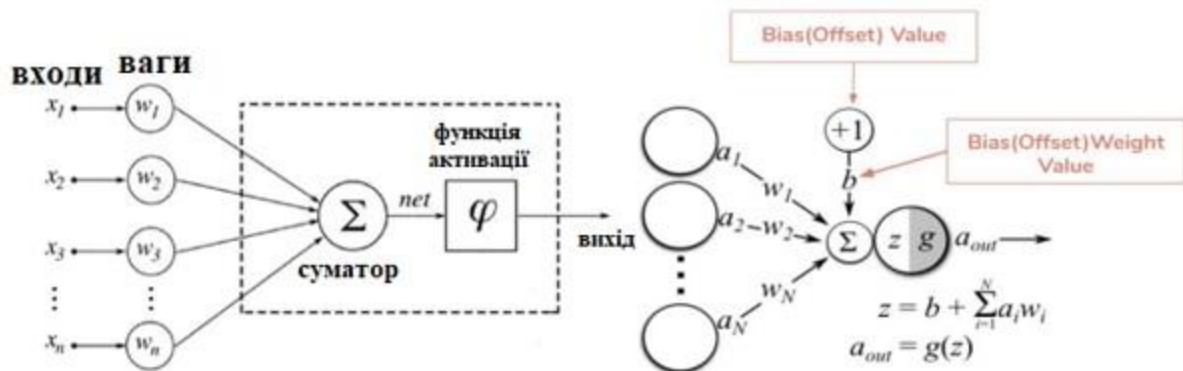


Рисунок 2.1 – Математична сутність штучного нейрону

Мережі прямого зв'язку є статичними в тому сенсі, що для даного входу вони виробляють набір вихідних значень, незалежних від попереднього стану мережі. Рекурентні мережі є динамічними, оскільки входи до нейронів у них модифікуються за рахунок зворотного зв'язку, що призводить до зміни стану мережі [51].

У загальному визначенні поняття «штучна нейронна мережа» охоплює ансамблі нейронів будь-якої структури, однак практичне застосування знайшли лише деякі з них. Це пояснюється тим, що архітектура ШНС є безпосередньо

пов'язаною з методом її навчання. Навіть різні етапи розвитку штучних нейронних систем визначалися появою нових архітектур мереж і спеціально розроблених для них методів навчання.

Штучний нейрон (або скорочено нейрон) є основним функціональним будівельним блоком, з якого побудовано багато штучних нейронних систем. Це жива модель нейрона, але лише в сенсі перетворень, які вона здійснює, а не в тому, як вона функціонує. Нейрони мають логічні, безперервні та спайкові моделі. Логічні моделі нейрона (зокрема, описуваний картою Відень формальний нейрон) активно досліджувалась в 60–70-х роках, але не отримали подальшого розвитку [52]. Імпульсні моделі ближчі до фізичну природу процесів, що відбуваються в нервовій клітині, проте їх теорія не так розвинена як у безперервних, і вони все ще не знаходять широкого застосування.

Адаптивний суматор обчислює скалярний добуток вектору вхідного сигналу на вектор параметрів. Адаптивним він називається через наявність вектору параметрів, що настроюються. Для багатьох задач корисно мати лінійну неоднорідну функцію з вихідним сигналом, для якої додається постійний єдиний вхідний сигнал.

Нелінійний конвертер сигналу – приймає скалярний вхідний сигнал і перетворює його відповідним чином.

Точки розгалуження використовуються для маршрутизації сигналу на кілька адрес.

Вона також отримує скалярний вхідний конвеєрних сигнал і передає його всім своїм виходів [53]. Формальний нейрон реалізує передавальну функцію (2.1):

$$\varphi(x) = \frac{1}{1 + e^{-N(x_1, x_2, \dots, x_n)}} = \varphi \left(\sum_{i=1}^n \alpha_i \cdot x_i + \alpha_0 \right) \quad (2.1)$$

Нейрональні шари можуть бути сформовані з формальних нейронів, які, у свою чергу, можуть бути об'єднані в багат шарові мережі. Нейрони вхідного рівня

отримують сигнали, перетворюють їх і передають нейронам прихованого шару через точки розгалуження. Далі наступний рівень запускається до вихідного рівня, сигналізуючи інтерпретатору та користувачеві. Кожен вихід нейрона будь-якого шару подається на вхід усіх нейронів наступного шару. Число нейронів в шарі [54] може бути будь-яким (рис. 2.2).

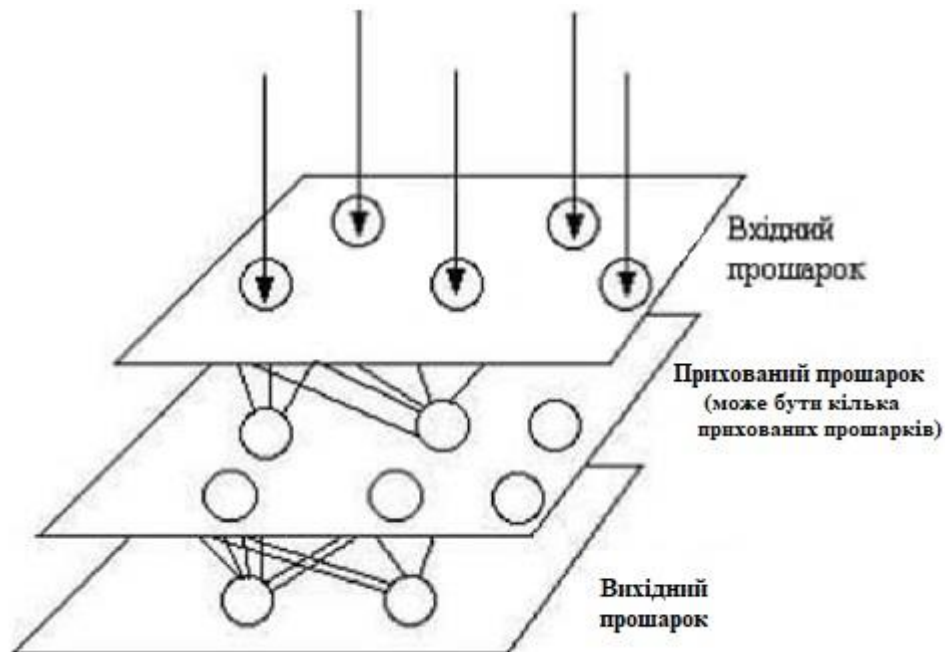


Рисунок 2.2 – Схематичне зображення штучної нейронної мережі

2.2 Математичний опис нейронної мережі NERF

Вперше у 1943 році нейрофізіолог Уоррен МакКаллох і математик Уолтер Пітс разом із моделлю нейронної мережі вперше запропонували математичну модель штучних нейронів [55]. Приблизно в той же час нейропсихолог Дональд Хебб зробив значний внесок у розвиток нейронних мереж, запропонувавши в 1949 році правила, що містяться в алгоритмах навчання для штучних нейронних мереж [56]. Робота цих вчених викликала першу хвилю інтересу до нейронних мереж, яка посилилася завдяки роботі Френка Розенблата (Frank Rosenblatt), Марвіна Мінскі (Marvin Minsky), Сеймур Пейперт (Seymour Papert), Бернарда Відроу (Bernard

Widrow), Марсіана Хоффа (Marcian Hoff).

З математичної точки зору нейронні мережі являють собою спосіб вирішення нелінійних завдань оптимізації. Кібернетика використовує теорію нейронних мереж у вирішенні завдань адаптивного управління, побудові алгоритмів для робототехніки. У програмуванні нейронна мережа один із способів вирішення проблеми ефективного паралелізму. Програмування нейронних мереж передбачає саме навчання мережі, а не написання програмного коду. Саме завдяки навчанню мереж здатна виявляти залежність між даними (вхідними та вихідними) узагальнювати, спрощувати результати, використовувати знання для розбиття складних завдань на більш прості [57].

ANN (artificial neural networks) – це математична модель функціонування традиційних для живих організмів нейромереж, які представляють собою мережі нервових клітин. Як і в біологічному аналогу, в штучних мережах основним елементом виступають нейрони, з'єднані між собою і утворюють шари, число яких може бути різним у залежності від складності нейромережі і її призначення.

Для опису математичної моделі даної нейронної мережі будемо використовувати такі позначення. Під $l \in [1, L]$ будемо розуміти розглядуваний наразі шар нейронної мережі, де $L = 2a + 2, a \in Z^+$ – кількість шарів в мережі. За позначимо N^l кількість карт ознак шару l , а за f_1 – функцію активації розглядуваного шару l . Також, під змінною y^l_n , будемо розуміти n -ну карту ознак шару l .

Математична модель шару згортки: Розглянемо шар згортки l . В архітектурі згорткової нейронної мережі l приймає непарні значення, тобто (2.2):

$$l = 1, 3, \dots, 2a + 1. \quad (2.2)$$

Тоді, для карти ознак n матиме місце наступне:

$w = \{w_{m, n}^1(i, j)\}$ – згортка, застосовувана до карти ознак m шару $(l - 1)$, на l шарі l з картою ознак n .

b^l_n – порогові ознаки, приєднані до карти ознак n на шарі l .

V_n^l – список усіх рівнів шару $(l - 1)$, які з'єднуються з картою ознак n шару l .

Таким чином карта ознак n шару згортки l буде обчислюватися [58] за формулою (2.3):

$$y_n^l = f_l(\sum_{m \in V_n^l} y_m^{l-1} \otimes w_{m,n}^l + b_n^l), \quad (2.3)$$

де оператор \otimes означає математичну операцію двовимірної згортки. Припустимо що розмір вхідних карт ознак y_m^{l-1} дорівнює $H^{(l-1)} \times W^{(l-1)}$, а розмір застосовуваної до них згортки $w_{m,n}^l - r^l \times c^l$, тоді розмір вихідної карти ознак y_n^l обчислюється (рис. 2.3):

$$(H^{(l-1)} - r^l + 1) \times (W^{(l-1)} - r^l + 1), \quad (2.4)$$

де $H^{(l-1)}, W^{(l-1)}$ ширина та висота вхідної карти ознак.

$w_{m,n}^l$ – згортка застосовувана до вхідної карти ознак m шару $(l - 1)$, $r^l \times c^l$ – розміри (ширина та висота) ядра матриці згортки (рис. 2.3).

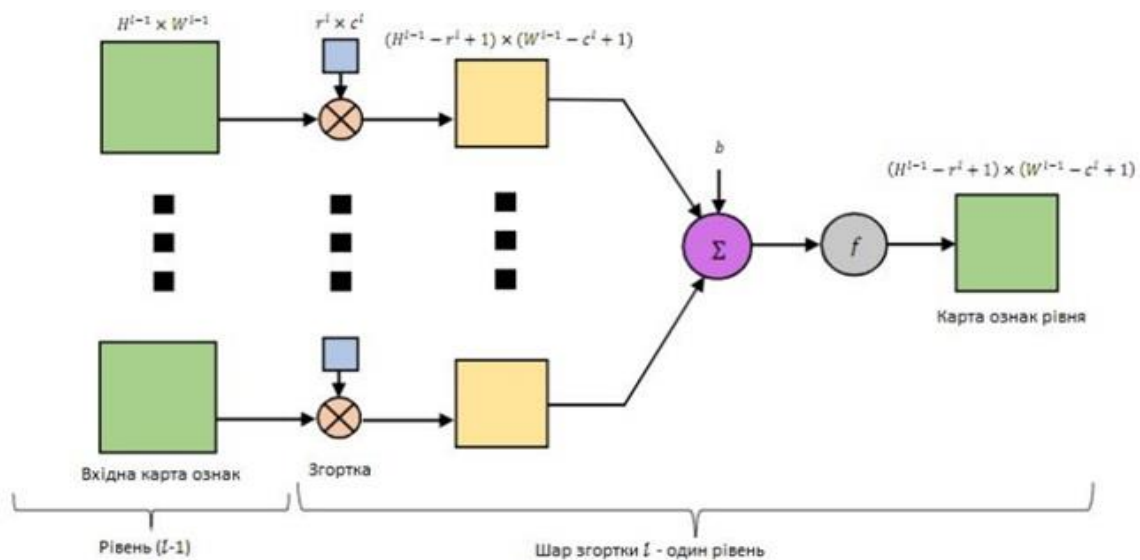


Рисунок 2.3 – Схема шару згортки

З математичної точки зору зображення є двовимірною матрицею значень яскравості. Однак при переході від однієї його частини до іншої, навіть такі статистики першого порядку, як гістограми значно змінюються. Існує набір класичних функцій, що застосовуються в вейвлет перетвореннях: вейвлет Хаара, вейвлет Морлі, вейвлет Добеши і т.д.

Хорошим прикладом застосування вейвлет аналізу є завдання пошуку відблиску в зіниці ока, де вейвлетом є сам відблиск. В основі вейвлетів лежить кореляція, яка може застосовуватися як в сукупності з іншими методами, так і самотійно. При розпізнаванні образу в зображенні це незамінний інструмент. Іншим не менш цікавим класом фільтрації є фільтрація функцій. Вона дозволяє на простому зображенні знайти безліч шматочків найпростіших функцій [59]: пряма, парабола та інші.

Neural Radiance Fields (NeRF) – метод глибокого навчання для генерації зображень нових будь-яких ракурсів для конкретного фотосету. Сама нейромережа навчається на обмеженій кількості фотографій. У нашому випадку це всього лише 22 фотографії особи з різних ракурсів. Таким чином, наша сцена досить розріджена, рішення задачі побудови зображень з нового ракурсу не є очевидним.

Основний підхід для створення нових зображень за допомогою NeRF. Даний метод був представлений на конференції ECCV навесні 2020. Сама стаття була удостоєна нагороди Best Paper Honorable Mention.

З тих пір з'явилися нові підходи для вирішення подібного завдання, але, тим не менш, даний підхід досі актуальний і показує Sota результати для генерації нових зображень [60].

Саму нейронну мережу можна уявити як складну функцію багатьох змінних. На вхід цієї функції подаються положення точки простору (x, y, z) , в якій передбачається знаходиться об'єкт, і вектор напрямку, заданий кутам θ та φ .

$$(x, y, z, \theta, \varphi) \rightarrow F_{\Theta} \rightarrow (RGB\sigma) . \quad (2.5)$$

Функція f повертає 4 числа σ або щільність, це деяке число, значення якого

дає нам зрозуміти, чи є якийсь об'єкт у зазначеній точці простору чи ні. Можна вважати це деяким аналогом прозорості зі значеннями від 0 до 1. RGB ж показують значення кольору в зазначеній точці простору (x, y, z) .

Ця функція F є повноцінною нейронною мережею MLP. Спочатку на вхід подаються координати (x, y, z) , які обробляються 8 повнозв'язними шарами (з функціями активації ReLU після кожного шару, що мають 256 каналів). Вихід з цієї частини мережі є щільність σ та 256 розмірний вектор, який потім об'єднується з вектором напрямку θ, φ і подається одному повнозв'язному шару (з ReLU активацією), щоб отримати значення RGB.

Ви запитаете, навіщо нам напрямки θ, φ , адже досить знати положення точки в просторі (x, y, z) . Додаткова залежність від напрямку «погляду» враховується для того, щоб більш точно моделювати ефекти відображення і освітленості і консистентно їх узагальнювати в кожній точці простору.

Чому сітка називається Neural Radiance Fields? Справа в тому, що через кожну точку простору проходять промені, що задаються радіальними координатами θ та φ . На кожному промені задається деякий розподіл $\sigma(x)$ для того, щоб моделювати безперервний обсяг, volume, всередині простору і показувати, в який момент виникає перешкода на шляху променя. Таким чином навчаємий розподіл всередині обсягу [61] – пряма альтернатива вокселізації (рис 2.3).

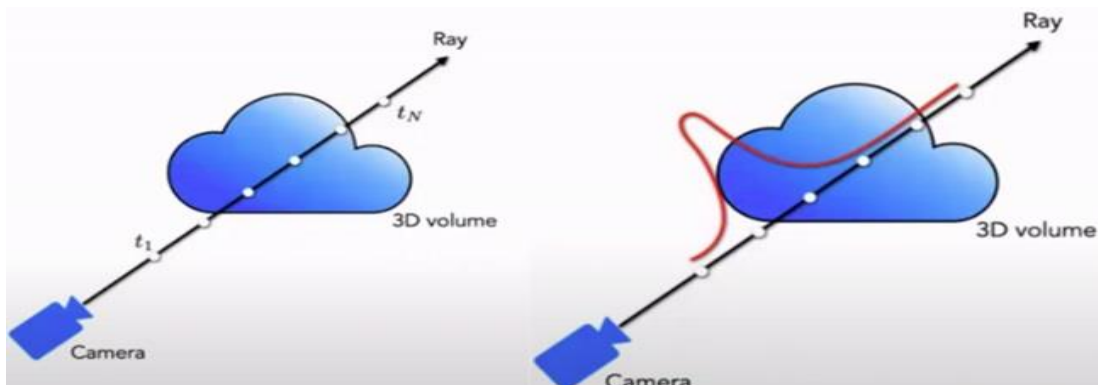


Рисунок 2.3 – На промені семплексується N точок, щоб змоделювати розподіл щільності на конкретному напрямку

Параметри t_1 та t_N задають грубі межі об'єкту. Дані значення можуть бути отримані за допомогою Multiview Stereo додатків COLMAP або Agisoft Metashape.

По суті, ми маємо модель для відновлення кольору і щільності розподілу на промені. Між значеннями t_1 та t_N ми рівномірно семплуємо N точок. Значення кольору на промені $r(t) = o + td$ у зазначеній точці обчислюється наближено:

$$\hat{C}(r) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i, \text{ where } T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j), \quad (2.6)$$

де $\delta_i = t_{i+1} - t_i$;

T_i – показує як багато світла було поглинуто перед точкою s індексом i на промені, тобто по суті це ймовірність, що промінь пройде від точки T_N до точки t без перешкод;

$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ призводить функцію $\sigma(x)$ до виду імовірнісного розподілу.

Зауважимо, що величина $\hat{C}(r)$ диференціюється за параметрам MLP, і ми можемо використовувати градієнтні методи, щоб оптимізувати NeRF. Щоб зробити семплювання на промені більш оптимальним, автори пропонують наступну евристику-оптимізацію відразу двох мереж, одна з яких називається Coarse, а інша Fine. Перша мережа отримує перше грубе наближення кольору в заданій точці простору на промені, на якому рівномірно насемплюванні N_c точок:

$$\hat{C}(r) = \sum_{i=1}^{N_c} w_i c_i, w_i = T_i (1 - \exp(-\sigma_i \delta_i)) \quad (2.7)$$

Ваги, з якими підсумовуються значення кольору RGB на промені, після нормалізації задають деякий розподіл, з якого можна додатково семплувати N_f точок на промені в місцях, де ймовірність знаходження об'єкту висока, і тим самим отримувати більш точні значення щільності σ та кольору RGB.

На цій доповненій множині з $N_c + N_f$ точок обчислюється фінальне значення кольору Fine сіткою. Це вимагає значно більше обчислювальних ресурсів, але дає більш точну оцінку розподілу.

Що стосується навчання NeRF, то це відбувається наступним чином. Вибирається зображення з навчальної вибірки, і для нього семплюється задане число променів, що проходять через цю фотографію. На кожному промені оцінюється значення кольору $\hat{C}(r)$ – колір там, де починається об'єкт (рис. 2.5 та 2.6).

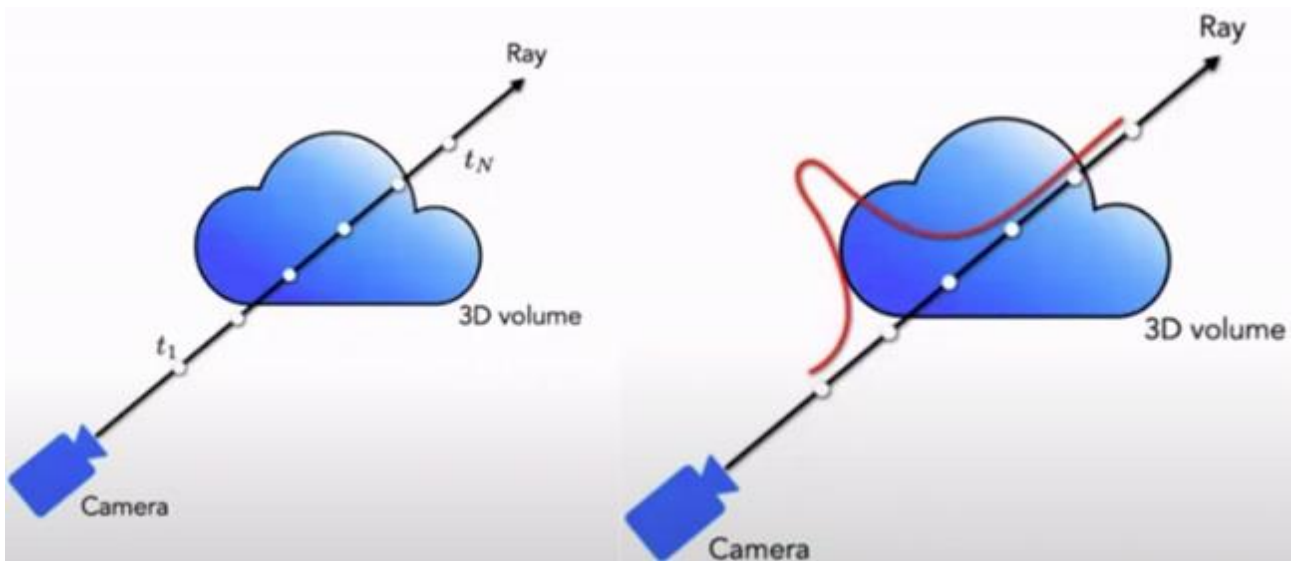


Рисунок 2.5 – Уточнення розподілу на промені після додаткового семплювання для Fine сітки

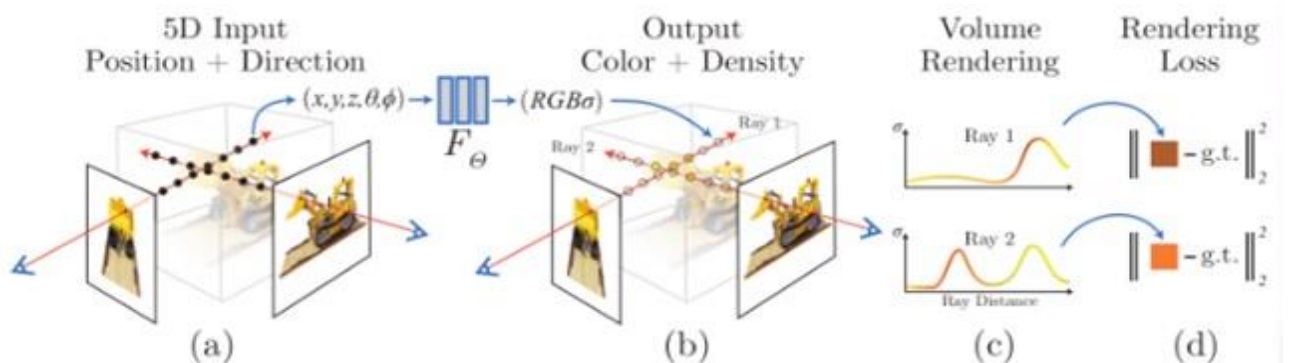


Рисунок 2.6 – Схематичне зображення процесу навчання мережі NeRF

В результаті обчислюється функція втрат, це така функція, яка відображує подію або значення однієї чи багатьох величин, число, яке інтуїтивно представляє деякі «витрати», протилежна функції користі, пов'язані з цією подією. Задача оптимізації намагається функцію втрат мінімізувати. Оптимізована за допомогою Adam. Одним з бонусів моделювання розподілу щільності на промені є можливість розуміти розташування об'єктів в просторі. Таким чином, NeRF можна використовувати для отримання досить якісних карт глибин зображень:

$$L = \sum_{\mathbf{r} \in R} [\|\hat{C}_c(\mathbf{r}) - C(\mathbf{r})\|_2^2 + \|\hat{C}_f(\mathbf{r}) - C(\mathbf{r})\|_2^2]. \quad (2.8)$$

РОЗДІЛ 3 ПРАКТИЧНЕ ВИКОРИСТАННЯ МЕРЕЖІ NERF ДЛЯ СТВОРЕННЯ 3D ЗОБРАЖЕНЬ

3.1 Існуючі переваги та недоліки нейронної мережі NERF

Розпізнавання образів (РО) – це процес розділу однотипних образів на класи за допомогою відповідних методів та засобів для виділення, ідентифікації, вимірювання та інтелектуальної обробки часто неповних, нечітких, непрямих, перекручених та суперечливих ознак образів.

Засоби, які використовуються при РО розподіляються на три групи:

- технічні датчики (зображення, температури, розміру, кольору, швидкості руху, тиску, ваги, магнітного поля і т. п.), які дозволяють виділити і «зважити» суттєві первинні ознаки образу;
- технічні засоби інтелектуальної обробки (переробки, порівняння, об'єднання, переоформлення, накопичення) інформації, в результаті чого отримують вторинні ознаки, у просторі яких здійснюється, власне, класифікація;
- математичне та програмне забезпечення у складі методів, вирішальних правил, обмежень, алгоритмів, програм, які використовуються для визначення класу образу.

Реальна розпізнавальна система звичайно складається з блоку, який виробляє ознаки (рецептор, у якому первинні ознаки переробляються у вторинні ознаки); блоку прийняття рішень (класифікатора), який розподіляє образи на класи; виконавчого пристрою, який виконує відповідну дію.

На самому початку проаналізуємо існуючі на даний час нейронні системи для створення зображення.

Alex Net. У 2012 році на світ з'явилася Архітектура AlexNet, створена Алексом Крижевским. Це була важлива робота для розвитку СНР, в якій автори вперше на той момент використовували глибокі згорткові нейромережі із загальною глибиною в 8 шарів (5 згорткових і 3 повнозв'язних шар). Архітектура

Крижевського чимось може нагадувати мережу LeNet, але Alex Net має більше фільтрів та вкладених згорткових шарів. Мережа включала в себе, крім згорток і пудингів, аугментацію даних, дропаут і функцію активації ReLU. Мережа навчалася на двох графічних процесорах протягом тижня. Архітектура AlexNet представлена на рисунку 3.1.



Рисунок 3.1 – Приклади аугментації з використанням Alex Net

Дана мережа в 2012 році виграла конкурс з розпізнавання зображень ImageNet з великим відривом (з кількістю помилок 15,3%, а у другого місця 26,2%) (рис 3.2).

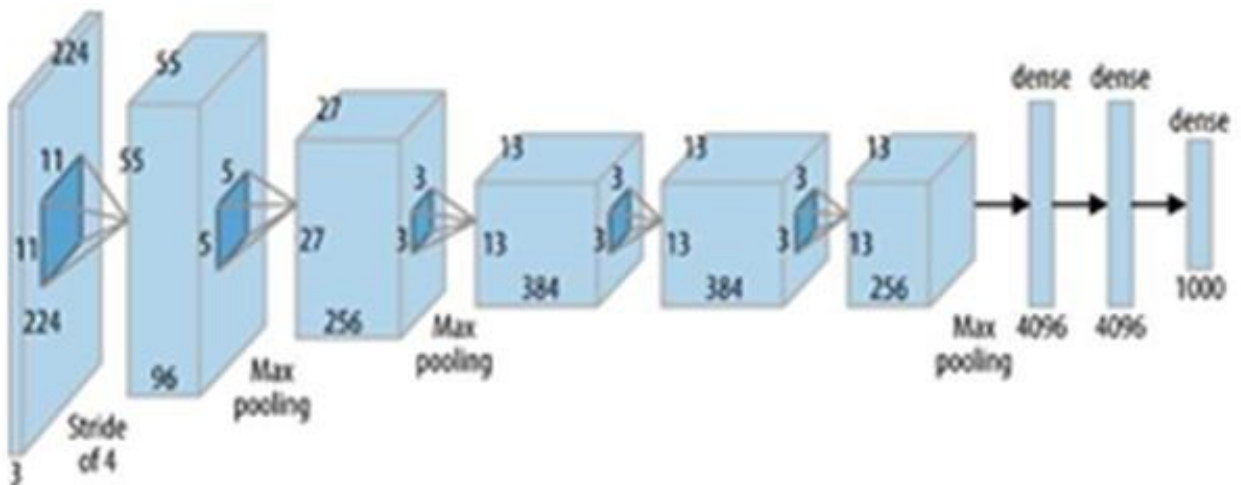


Рисунок 3.2 – Архітектура AlexNet

ResNet. У 2015 році відбулася чергова революція – на світ з'явилася глибока згортова нейронна мережа ResNet. Майже всі нейронні мережі як метод навчання використовують метод зворотного поширення помилки. При зворотному проході застосовується градієнтний спуск з метою зменшення функції помилки. При збільшенні кількості шарів, тобто глибини нейромережі, часта проблема – це загасання градієнта при зворотному проході.

Тобто, якщо градієнт має невелике значення в кінці мережі, то він може прийняти дуже мале значення, коли досягне початку мережі. Для прикладу $0,5^{50} = 8 \cdot 10^{-16}$ Мережа перестає навчатися, тобто не має сенсу просто збільшувати кількість шарів. В основі даної архітектури лежать блоки з сполуками швидкого доступу (англ. shortcut connections), зображені на рисунку 3.3.

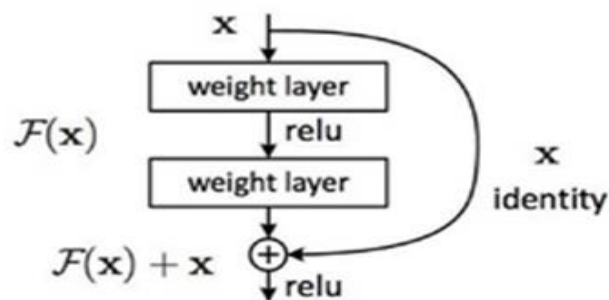


Рисунок 3.3 – Схематичне зображення блоку з швидких з'єднань

Схему даного блоку можна пояснити так: тепер цільова функція має вигляд $G(x) = F(x) + x$, такі блоки додають тотожне перетворення в мережу. Отже, при зворотному проході отримуємо

$$\frac{dG(x)}{dx} = \frac{dF(x)}{dx} + \frac{dx}{dx} = \frac{dF(x)}{dx} + 1. \quad (3.1)$$

Тепер загасання градієнта не відбудеться, тому що доданок одиниця дозволяє» протікати " градієнту аж до перших шарів.

Нижче на рисунку 3.4 представлена мережа ResNet18-одна з варіацій архітектури ResNet з 18 шарами і застосування блоків з shortcut connections в мережі (рис. 3.4).

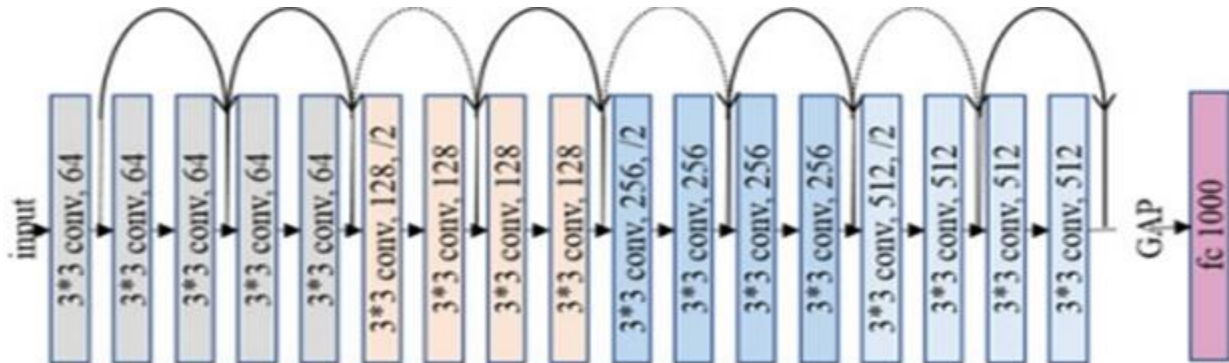


Рисунок 3.4 – Схема сети ResNet18

PyTorch – сучасний фреймворк глибокого навчання для мови Python. Pytorch дозволяє працювати з нейронними мережами. Для реалізації навчання нейромережі необхідно написати код тренування моделі.

Створення моделі в Pytorch не вимагає великих витрат часу. Також важливими перевагами є легка інтеграція з графічним процесором для обчислень, наявність попередньо навчених моделей і готових модульних частин, які легко комбінувати.

Код навчання моделі виглядає так:

```

1 loss_train_history = []
2 loss_val_history = []
3 acc_train_history = []
4 acc_val_history = []
5
6 has_gpu = torch.cuda.is_available ()
7 dataloaders = {'train ': train_dataloader , 'val ': val_dataloader }
8 def train_model (model, optimizer, criterion, num_epochs =25):
9 best_acc = 0.0
10
11 for epoch in range (num_epochs):
12 for phase in ['train ', 'val ']:
13 if phase == 'train ':
14 model.train (True ) # model in training mode
15 else:
16 model.train (False ) # model in evaluate mode
17 current_loss = 0.0
18 current_corrects = 0
19

```



```

20 data_loader = dataloaders [phase]
21 for data in data_loader:
22 # using dataset
23 inputs, labels = data
24 if has_gpu:
25 inputs = Variable (inputs. cuda () ,)
26 labels = Variable (labels. cuda () )
27 else:
28 inputs = Variable (inputs)
29 labels = Variable (labels)
30
31 optimizer. zero_grad ()
32 # forward pass
33 outputs = model (inputs)
34 _, preds = torch. max (outputs. data, 1)
35 loss = criterion (outputs, labels)
36 # backward pass
37 if phase == 'train ':
38 loss. backward ()
39 optimizer. step ()
40 # data for graphics
41 current_loss += loss. data [0] * inputs. size (0)
42 current_corrects += torch. sum (preds == labels. data)
43
44 epoch_loss = current_loss / len (data_loader. dataset)
45 epoch_acc = current_corrects / len (data_loader. dataset)
46 if phase == 'train ':
47 loss_train_history. append (epoch_loss)
48 acc_train_history. append (epoch_acc)
49 else:
50 loss_val_history. append (epoch_loss)
51 acc_val_history. append (epoch_acc)
52
53 if phase == 'val ' and epoch_acc > best_acc:
54 best_acc = epoch_acc
55
56 return model

```

Keras – це API високого рівня нейронної мережі, написаний на Python, здатний

працювати поверх TensorFlow, CNTK або Theano. Він розроблений з акцентом на швидке експериментування. Здатність переходити від ідеї до результату з найкоротшою затримкою є ключем до якісного дослідження. Keras спочатку був розроблений для дослідників, що дозволяє їм швидко проводити експерименти.

Keras має такі ключові особливості:

- API дозволяє виконувати один і той же код на CPU або GPU;
- має дружній для користувачів API, що спрощує розробку прототипів моделей глибокого навчання;
- включає в себе підтримку згортальних мереж (для розпізнавання образів), рекурентних мереж (для обробки послідовностей) і всіляких можливих їх комбінацій;

- також включає в себе підтримку довільних мережевих архітектур: моделей з множинними входами або виходами, спільне використання шарів, спільне використання моделей і т.д.

Це значить, що API Keras підходить для створення майже будь-якої моделі глибокого навчання. Фреймворк Keras поширюється на умовах безкоштовної ліцензії, вільний для використання в комерційних проектах. Він сумісний з будь-якою версією Python, від 2.7 до 3.8.

Існує понад 200 000 користувачів Keras, починаючи від академічних дослідників та інженерів стартапів і великих корпорацій до аспірантів і любителів. Keras використовують Google, Netflix, Uber, CERN, Yelp, Square і сотні стартапів, що вирішують широкий спектр завдань. Крім того, Keras — це бібліотека рівня моделі, яка надає високорівневі будівельні блоки для створення моделей глибокого навчання. Він не реалізує операції низького рівня, такі як операції з тензорами та диференціюванням - для цього використовується спеціалізована та оптимізована бібліотека підтримки тензорів. У той же час Keras не покладається на одну бібліотеку підтримки тензорів, а використовує модульний підхід, тобто кілька різних бібліотек низького рівня можна підключити до фреймворку Keras.

Наразі підтримуються три такі бібліотеки: TensorFlow, Theano та Microsoft Cognitive Toolkit (CNTK). Будь-який код, який використовує Keras, можна запускати з будь-якої з цих бібліотек, не змінюючи нічого у вашому коді: ви можете швидко перемикатися між ними під час розробки, що часто корисно, наприклад, якщо одна з бібліотек знаходиться в цьому конкретному завданні демонструє кращу продуктивність при вирішенні наступних завдань.

NumPy — це бібліотека для мови програмування Python, яка забезпечує підтримку великих багатовимірних масивів і матриць. Використовуйте низку передових математичних функцій для маніпулювання цими масивами.

Головною особливістю NumPy є його структура даних для n-вимірних масивів. Ці масиви суворі для пам'яті. На відміну від вбудованої структури даних списку Python, усі елементи цього масиву мають бути одного типу. Такі масиви також можуть переглядати буфери пам'яті, виділені інтерпретатору CPython

розширеннями C/C++, Cython і Fortran, без копіювання даних, таким чином забезпечуючи певний ступінь сумісності з існуючими числовими бібліотеками.

NeRF – це вкрай значуща робота в області генерації нових фотореалістичних зображень з різних ракурсів. Вона викликала феномен в області 3D Deep Learning, званий NeRF Explosion – протягом року після публікації з’явилося багато статей з видатними результатами, що використовують і доповнюють ідеї NeRF для Novel View Synthesis. У нейронній мережі NeRF багато переваг:

- безперервне і консистентне відновлення сцени;
- Можливість роботи в сценарії, коли кількість фотографій вкрай обмежена;
- висока деталізованість генерованих зображень в порівнянні із стандартними методами рендеринга зображень з текстурою;

- в даній мережі можна симулювати view-dependent відображення (спекулярні), при цьому потрібно всього лише пара десятків фото з різних ракурсів. Тобто це вже не чисто фотограмметрія, а відновлення radiance-field, а цілісного для novel view. І тому можна розрахувати якісно повноцінну фотограмметрію за допомогою оцінки weighted-depth (вже є такі роботи про nerf);

- можна розрахувати за допомогою чисельних методів оптимізації albedo, metallic, roughness параметри.

На даний час існують дослідження, що показують динамічну підтримку NeRF з допомогою трансформації. А також підтримують відновлення radiance-field по фотографіях з великою кількістю непотрібних динамічних об’єктів поверх потрібних нам сцен (наприклад фото туристичних місць) або при різній освітленості. Тобто алгоритм з використанням matting і додаткових перетворень досить стабільний. А це може дозволити проводити експорт анімації.

Основні недоліки нейронної мережі NeRF такі:

- Кількість фотографій для навчання не може бути занадто малим – має бути помітним перетин областей на зображеннях.

- Фон на фотографіях викликає сильне перенавчання нейромережі і вносить значний шум в генеровану сцену. Для вирішення цієї проблеми ми використовували видалення фону за допомогою згаданої background matting нейромережі. На

нижньому рисунку та в додатку роботи наведено приклади очистки фону. Зліва фон був залишений у фотографіях навчальної вибірки, а праворуч фон у навчальній вибірці було видалено (рис 3.5).



Рисунок 3.5 – Демонстрація впливу фону на якість зображення

Для навчання потрібні значні обчислювальні (GPU, CPU) та тимчасові ресурси. Для прискорення експериментів розумно використовувати версію NeRF від Google Research, написану з елементами AJAX і розпаралелювання на декількох GPU і TPU. З нашого досвіду, це значно скорочує як час навчання нейромережі, так і фінальний час генерації нових зображень.

Досліджувана мережа тренується тільки під конкретну сцену. Через архітектуру NeRF можна навчити лише конкретній сцені. Якщо ви хочете відновити безперервне представлення нової сцени, вам буде потрібно заново тренувати цю нейромережу на новому наборі фотографій. У свіжих статтях представлені способи отримання novel views на сценах, яких не було в навчальній вибірці. Але про це ми поговоримо в наших наступних публікаціях.

NeRF фіксує освітлення та фотометричну постобробку в низькорозмірному прихованому просторі вбудовування. Інтерполяція між двома вбудовуваннями плавно фіксує варіації зовнішнього вигляду, не впливаючи на 3D-геометрію. NeRF не вимагає, щоб кожне зображення перекривалося на 50%. Тож хоча перекриття все

ще потрібні, штучний інтелект допомагає, заповнюючи невеликі прогалини. Поряд з наявністю недоліків використання досліджуваної нейронної мережі є корисним для фотореалістичної 3D реконструкції людей (рис 3.6).



Рисунок 3.6 – Приклад створення зображення з допомогою NeRF

3.2 Практичне використання мережі NERF для створення зображень

Проаналізуємо основні етапи створення зображень мережею NERF.

Підготовка даних. Оригінальні фотографії мають складний фон і, як ми покажемо, це вносить значний шум у модель. Для вирішення цієї проблеми ми застосовуємо background matting за допомогою нейромережі, яка вкрай точно прибирає фон з фотографії (рис 3.7).



Рисунок 3.7 – Вихідні фотографія До та після видалення фону

Дані для навчання. Ми маємо обмежене число фотографій особи з різних ракурсів. Зокрема, ми маємо 22 фотографії обличчя людини високої роздільної здатності 6000x4000 з різних ракурсів, отриманих за допомогою камер для фотограмметрії

Геометрія і риси обличчя збережені, освітлення змінюється безперервно між усіма новими зображеннями. Також вона непогано впоралася з рендерингом волосся і очей – досить складним завданням в області 3D Computer Vision. Так, на перших трьох малюнках вийшло вивчити зачіску з точністю до волосків.

Набір фотографій особи, отриманих за допомогою камер для фотограмметрії приведено на рисунку 3.8. При цьому, було застосовано background matting для видалення фону з фотографій.



Рисунок 3.8 – Набір фотографій особи, отриманих за допомогою камер для фотограмметрії

В подальшому для моделі важливо оцінити за наявним набором фотографій екстринсики та інтринсики камер, які називаються параметрами зовнішнього і

внутрішнього калібрування для визначення положення камер в просторі, фокусної відстані, principal points (рис. 3.9).

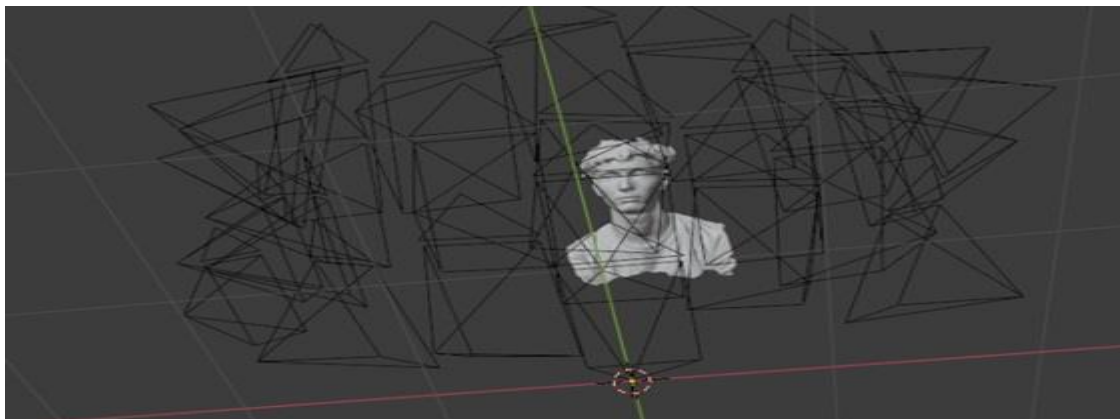


Рисунок 3.9 – Сетап сцени, що вивантажений в Blender

Це можна зробити як і в безкоштовному пакеті COLMAP, так і в платному Agisoft Metashape. Використання Agisoft Metashape представлено на рис. 3.10:



Рисунок 3.10 – Обличчя з нових ракурсів

У twin3d ми маємо можливість застосовувати і тестувати State-of-the-art

підходи машинного і глибокого навчання на наших даних – фотографіях вкрай високого розрішення і фотореалістичних 3D моделях, отриманих за допомогою сетапа для фотограмметрії. Нам було цікавим випробувати NeRF на наших high-resolution фотосетах. Ми протестували алгоритм на різних сценах обличчя, кожна з яких мала лише 22 фотографії з різних ракурсів. Отримані результати представлено на рис. 3.10.

На рис. 3.10 представлені обличчя з нових ракурсів, які були згенеровані NeRF. Важливо відзначити, що на оригінальних фотографіях освітлення трохи відрізнялося. NeRF вивчив світло саме таким, яким він був на кожній фотографії, і рівномірно інтерполював його між novel views.

Таким чином, NeRF показав вельми цікаві і вражаючі результати на фотосетах для 3D фотограмметрії.

ВИСНОВКИ

1. На початку дослідження наголошено на тому, що графічні системи розпізнавання інформації відіграють важливу роль і є основним компонентом обчислювальних пристроїв, які використовуються в різних сферах людської діяльності. Такі системи можуть істотно спростити вирішення повсякденних завдань, пов'язаних з обробкою графічної інформації, що надає їм особливе місце в галузі.

2. Розкрито сутність поняття «штучні нейронні мережі» та названо головні складові нейронної мережі. Описано особливості створення штучного нейрону та акцентовано увагу на тому, що потім тіло штучного нейрона підсумовує зважені вхідні дані, зміщення та обробляє суму за допомогою функції передачі. Основною невідомою змінною нашої моделі є її функція активації. Функція активації визначає властивості штучного нейрона і може бути будь-якою математичною функцією. Проаналізовано три методи навчання та наведено їхні переваги та недоліки. Огляд нейронних мереж, які використовуються для створення зображень.

3. В другій частині роботи описана архітектура штучних нейронних мереж і сказано, що за архітектурою зв'язків ШНС можуть бути згруповані в два класи: мережі прямого поширення, в яких графи не мають петель, і рекурентні мережі, або мережі з зворотними зв'язками. Адаптивний суматор обчислює скалярний добуток вектору вхідного сигналу на вектор параметрів. Адаптивним він називається через наявність вектору параметрів, що настроюються. Для багатьох задач корисно мати лінійну неоднорідну функцію з вихідним сигналом, для якої додається постійний одиничний вхідний сигнал. Нелінійний конвертер сигналу – приймає скалярний вхідний сигнал і перетворює його відповідним чином. Точки розгалуження використовуються для маршрутизації сигналу на кілька адрес. Він приймає скалярний вхідний сигнал і передає його на всі виходи.4. Зроблено математичний опис нейронної мережі NERF та сказано, що сама нейромережа навчається на обмеженій кількості фотографій і може бути використана для отримання досить

якісних карт глибин зображень.

5. В практичній частині дослідження перелічено головні переваги та недоліки нейронної мережі NeRF. Наголошено, що для навчання потрібні значні обчислювальні (GPU, CPU) та тимчасові ресурси. Для прискорення експериментів розумно використовувати версію NeRF від Google Research, написану з елементами AJAX і розпаралелювання на декількох GPU і TPU. З нашого досвіду, це значно скорочує як час навчання нейромережі, так і фінальний час генерації нових зображень.

6. Крім того, досліджувана мережа тренується тільки під конкретну сцену. Через архітектуру NeRF можна навчити лише конкретній сцені. Якщо ви хочете відновити безперервне представлення нової сцени, вам буде потрібно заново тренувати цю нейромережу на новому наборі фотографій. У свіжих статтях представлені способи отримання novel views на сценах, яких не було в навчальній вибірці. Але про це ми поговоримо в наших наступних публікаціях.

Наведено приклад створення якісних зображень з допомогою NeRF при наведеному програмний код застосунку.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Адаменко В. О., Мірських Г. О. Штучні нейронні мережі в задачах реалізації матеріальних об'єктів частина 2. Особливості проектування та застосування. Вісник Національного технічного університету України «КПІ». Серія – Радіотехніка. 2012. 340 с.
2. Аль Равашдех Лейт Ахмед Мустафа, Руженцев І. В. Оцінка похибок динамічних нейронних мереж для вимірювальних систем. *Метрологія та прилади*. 2018. 242 с.
3. Андреас М. Введення в машинне навчання за допомогою Python. Керівництво для фахівців по роботі з даними. М.: Альфакнига, 2017. 487 с.
4. Бардаченко В. Ф. Перспективи застосування імпульсних нейронних мереж з таймерним представленням інформації для розпізнавання динамічних образів. Вінниця: УНІВЕРСУМ-Вінниця, 2013. 267 с.
5. Бенгфорт Б. Прикладний аналіз текстових даних на Python. Машинне навчання та створення додатків обробки природної мови. Санкт-Петербург: Пітер, 2019. 368 с.
6. Бойчук В. О. Сучасні штучні нейронні мережі та підходи до їх моделювання. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2014. 519 с.
7. Бурлесєв О., Василенко О. Ефективність використання штучних нейронних мереж в економіці. *Економіка та суспільство*. 2021. № 3. URL: <https://doi.org/10.32782/2524-0072/2021-31-27>. (дата звернення: 04.12.2022).
8. Ващишин Л. В. Використання вейвлет-нейронних мереж для виявлення дефектів залізничних рейок. *Сучасні комп'ютерні інформаційні технології (АСІТ'2014)*. Тернопіль: ТНЕУ, 2014. 643 с.
9. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення [Електронний ресурс]. URL: <http://item.comp-sc.if.ua/2017/Sineglazov.pdf>. (дата звернення: 04.12.2022).

10. Доманецька І. М., Федусенко О. В. Нейромережеві технології опрацювання природномовних текстів в адаптивних системах навчання. *Штучний інтелект*, 2017, № 3–4. С. 24–32.
11. Жуковський В. В., Шатний С. В., Жуковська Н. А. Нейронна мережа для розпізнавання та класифікації картографічних зображень ґрунтових масивів. *Науковий вісник НЛТУ України*. 2020, т. 30, № 5. С. 100–104.
12. Ізонін І. В. Метод збільшення роздільної здатності зображень на основі штучних нейронних мереж. *Вісник Львівського державного університету безпеки життєдіяльності*. 2018. № 11. С. 47–56.
13. Казарян А. Г., Теслюк В. М, Коваль В. Я. Використання функції розпізнавання облич для контролю доступу користувачів та автоматизованого управління налаштувань приладів «розумного» будинку. *Моделювання та інформаційні технології*. 2018. Вип. 83. С. 180–185.
14. Калініна І. В. Використання генетичних алгоритмів в задачах оптимізації. *Міжвідомчий науково-технічний збірник*. 2015. № 1 (26). URL: <https://ela.kpi.ua/handle/123456789/16458>. (дата звернення: 04.12.2022).
15. Каллан, Р. Нейронні мережі: Короткий довідник. М.: Вільямс І. Д., 2017. 288 с.
16. Карпович А. В. Використання згорткових нейронних мереж для задачі класифікації текстів. *Міжнародний науковий журнал «Інтер-наука»*. 2018. № 14. URL: <https://doi.org/10.25313/2520-2057-2018-14-4105>.
17. Колесницький О. К. Аналітичний огляд апаратних реалізацій спайкових нейронних мереж. *Математичні машини і системи*. 2015. № 1. С. 3–19.
18. Комп'ютерні системи штучного інтелекту. Методичні вказівки до виконання лабораторних робіт студентами денної та заочної форми навчання. Укл.: Є. В. Мелешко. Кіровоград: КНТУ, 2016. С. 8–13.
19. Коротка Л. І., Науменко Н. Ю. Розробка нейромережевого програмного модуля планування траєкторії руху об'єкту на площині. *Інженерні та освітні технології*. № 2 (10), 2015. С. 28–35.
20. Косілова О. І. Солодовнікова Х. К. Права і свободи людини і громадянина

v. s. штучний інтелект: проблемні аспекти. *Інформація і право*. № 4 (35). 2020. С. 56–66.

21. Кривохата А. Г. Оптимізація згорткової нейронної мережі при розв'язанні задачі класифікації акустичних даних. *Інновації науки XXI століття: збірник наукових матеріалів XXXVI Міжнародної науково-практичної інтернет-конференції*. Вінниця, 2019. С. 23–27.

22. Кудін О. В., Кривохата А. Г. Генетичні алгоритми оптимізації ансамблів згорткових нейронних мереж. *Комп'ютерні науки, інформаційні технології та системи управління*. Івано-Франківськ, 2019. С. 28.

23. Лейт Ахмед Мустафа Аль Равашдех, Руженцев І. В. Оцінка похибок динамічних нейронних мереж у вимірювальних системах. *Метрологія та вимірювальна техніка*. Харків, 9–11 жовтня 2018 р. С. 190.

24. Лупей М. І., Міца О. В. Використання штучних нейронних мереж. *V Міжнародна науково-практична конференція «Обчислювальний інтелект (результати, проблеми, перспективи)*. 2019». Ужгород: ПП «Інватор». С. 253–254.

25. Масюк А. Р. Алгоритм інтелектуального вертикального хендоверу в гетерогенній мобільній мережі на основі хмарних обчислень. *Вісн. Нац. ун-ту «Львів. Політехніка»*. 2017. № 874. С. 110–121.

26. Надригайло Т. Ж., Молчанова К. А. Аналіз нейронних алгоритмів. *Математичне моделювання: електрон. наук. фахове вид.* 2015. URL: <http://www.dstu.dp.ua/Portal/Data/74/68/13 – st13.pdf>. (дата звернення: 04.12.2022).

27. Назаркевич М. А., Возний Я. В., Назаркевич Г. Я. Розроблення методу машинного навчання при біометричному захисті із новими методами фільтрації. *Кібербезпека: освіта, наука, техніка*. 2021. Т. 3, № 11. С. 16–30.

28. Нейронні мережі: їх застосування, робота [Електронний ресурс]. Режим доступу до ресурсу: <https://www.poznavayka.org/uk/nauka-i-tehnika2/neyronni-merezhi-yih-zastosuvannya-rob>. (дата звернення: 04.12.2022).

29. Пелешак Р. М., Литвин В. В. Розробка штучної нейронної мережі з осциляторними нейронами для розпізнавання спектральних образів. *Вісник Національного університету «Львівська політехніка»*. 2020. Вип. 7. С. 16–23.

30. Пустоваров В. В. Інформаційна технологія розробки бази знань на цифрових космічних та аерофотознімках при автоматизованому моніторингу міського середовища з використанням нейромереж та нечіткої логіки. XX науково-технічна конференція. *Створення та модернізація озброєння і військової техніки в сучасних умовах*. Чернігів, ДНДІВСОБТ, 2020. С. 121–122.

31. Пустоваров В. В. Архітектура нечіткої згорточної нейронної мережі для сегментації цифрових космічних та аерофотознімках при автоматизованому моніторингу міського середовища. *Polish journal of science*. №35, 2021. pp. 20–24.

32. Радутний О. Е. Кримінальна відповідальність штучного інтелекту. *Інформація і право*. № 2 (21). 2017. С. 124–132.

33. Руденко О. Г., Бодянський Є. В. Штучні нейронні мережі: Навчальний посібник. К.: Компанія СМІТ, 2006. 404 с.

34. Савка Н. Я., Спільчук В. М, Співак І. Я Проблеми ідентифікації штучних нейронних мереж з радіально-базисними функціями та можливі напрямки їх розв'язання. *Індуктивне моделювання складних систем: зб. наук. праць*. 2012. Вип. 2. С. 181–193.

35. Савка Н. Я. Етапи ідентифікації штучних нейронних мереж з радіально-базисними функціями. *Сучасні комп'ютерні інформаційні технології*. Тернопіль: ТНЕУ, 2018. С. 87.

36. Савченко В. А., Шаповаленко О. Д. Основні напрями застосування технологій штучного інтелекту у кібербезпеці. *Сучасний захист інформації*. 2020. № 4 (44). С. 6–11.

37. Сватюк Д. Р., Сватюк О. Р., Белей О. І. Застосування згорткових нейронних мереж для безпеки розпізнавання об'єктів у відеопотоці. *Кібербезпека: освіта, наука, техніка*. №4 (8), 2020. С. 97–112.

38. Сиротенко І. С., Щербина І. С. Аналіз ефективності використання нейронних мереж на прикладі багатопарового персептрону та мережі Кохонена. *Науковий журнал «Зв'язок»*. 2020. №5, с. 17–19.

39. Семенова О. О. Застосування нейронної мережі у процедурі вертикального хендоверу. *ІТКІ*. 2020. Вип. 49. С. 14–21.

40. Сторчак К. П., Тушич А. М. Інтелектуальний аналіз даних з використанням нейронних мереж. Науковий журнал «Зв'язок». 2018. №4, с. 17–19.
41. Сторчак К. П., Тушич А. М. Кластерний аналіз даних з використанням штучних нейронних мереж. *Науковий журнал «Зв'язок»*. 2018. №6. С. 36–38.
42. Тимофєєва А. Є. Кудін О. В. Кривохата А. Г. Лісняк А. О. Автоматичне анотування зображень за допомогою нейронних мереж. *Вчені записки Таврійського національного університету імені В. І. Вернадського*. 2019. Т. 30 (69). С. 214–220.
43. Тушич А. М., Сторчак К. П. Вимоги до інтелектуальних систем аналізу даних та їх класифікацій. *Науково-технічний журнал «Телекомунікаційні та інформаційні технології»*. 2019. №1, с. 31–36.
44. Цимбал Ю. В. Нейромережевий метод симетричного шифрування даних. *Вісник Національного університету «Львівська політехніка». Серія: Інформаційні системи та мережі*. 2018. № 901. С. 118–122.
45. Чабан Г., Кухтяк О. Застосування апарата теорії штучних нейронних мереж у задачах прикладної педагогіки закладів вищої освіти. *Український Педагогічний журнал*. 2020. № 1. С. 51–56. URL: <https://doi.org/10.32405/2411-1317-2020-1-51-56>.
46. Шефкін Б. В., Красюк І. В. Дослідження та впровадження нейронної мережі на основі TENSORFLOW. *Науковий журнал «Зв'язок»*. 2020. №6, с. 18–20.
47. About Keras. URL: <https://keras.io/about/> (дата звернення: 04.12.2022).
48. Ayala H. V. H. and L. dos Santos Coelho, Cascaded evolutionary algorithm for nonlinear system identification based on correlation functions and radial basis functions neural networks. *Mechanical Systems and Signal Processing*. 2016. Vol. 68. P. 378–393.
49. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, June 2017, pp. 84–90.
50. Husak V., Lozynska O., Karpov I., Peleshchak I., Chyrun S., Vysotskyi A. Information system for recommendation list formation of clothes style image selection according to user's needs based on NLP and chatbots. *CEUR Workshop Proceedings*, 2020. Vol. 2604. P. 788–818.

51. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. *ArXiv*. 2015. URL: <https://arxiv.org/pdf/1512.03385.pdf>. (дата звернення: 04.12.2022).
52. Krenker A. Introduction to the Artificial Neural Networks. *Artificial Neural Networks – Methodological Advances and Biomedical Applications*. 2011. 210 p.
53. Krizhevsky A. ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012, Pp. 1097–1105.
54. Larochelle H. Classification Using Discriminative Restricted Boltzmann Machines. *Proceedings of the 25th International Conference on Machine Learning / H. Larochelle, Y. Bengio.*, 2008. С. 536–543.
55. Nikonov D. E. Coupled-Oscillator Associative Memory Array Operation for Pattern Recognition. *Electrical and Computer Engineering Faculty Publications and Presentations*. 2015. Vol. 348, № 1. P. 85–93. URL: <https://core.ac.uk/download/pdf/37771867.pdf>. (дата звернення: 04.12.2022).
56. OpenCV About. URL: <https://opencv.org/about/>(дата звернення: 04.12.2022).)
57. Rida Sara Khan, Asad Ali Zardar, Zeeshan Bhatti. Artificial Intelligence based Smart Doctor using Decision Tree Algorithm. *Journal of Information & Communication Technology – JICT*. 2017. Vol. 11, Issue. 2. P. 1–5.
58. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 2014. № 15. P. 1929–1958.
59. Sun T., Pei H. Robust adaptive neural network control for environmental boundary tracking by mobile robots. *International Journal of Robust and Nonlinear Control*. 2016. Vol. 23, 123–136.
60. Fukuda T. Efficient Knowledge Distillation from an Ensemble of Teachers. *Proc. Interspeech 2017*, August 20–24, Stockholm, Sweden, 2017. P. 3697–3701.
61. Yoon Kim Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) – Doha, Qatar*. October 25–29, 2014. P. 1746–1751.