

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

БОДНАР Андрій Русланович

**Алгоритми розпізнавання гістологічних
зображень на основі згорткових нейронних
мереж / Histological Image Recognition Algorithms
Based on Convolutional Neural Networks**

спеціальність: 123 - Комп'ютерна інженерія
магістерська програма - Комп'ютерна інженерія

Магістерська робота

Виконав студент групи КІм-21
А. Р. Боднар




Науковий керівник:
д.т.н., професор, О. М.
Березький



Магістерську роботу допущено
до захисту:

31 " 01 " 2018 р.

Завідувач кафедри

 О. М. Березький

ТЕРНОПІЛЬ - 2018

Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії
Освітній ступінь «магістр»
Спеціальність 123 - комп'ютерна інженерія
Магістерська програма - Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

завідувач кафедри
д.т.н., проф. Березький О.М.

“14” 10 2018 р.

ЗАВДАННЯ НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Боднара Андрія Руслановича

1. Тема магістерської роботи «Алгоритми розпізнавання гістологічних зображень на основі згорткових нейронних мереж / Histological Image Recognition Algorithms Based on Convolutional Neural Networks»

Керівник роботи д.т.н., проф. О.М. Березький

затверджені наказом по університету від 11 жовтня 2016 р. № 669.

2. Строк подання студентом роботи «15» січня 2018 року.

3. Вихідні дані до магістерської роботи

Об'єкт дослідження – процес дослідження гістологічних зображень раку молочної залози.

Предмет дослідження – методи та алгоритми класифікації зображень, штучні нейронні мережі.

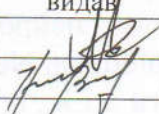

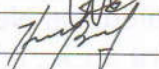
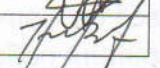
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- Привести опис і характеристики гістологічних зображень передракових станів молочної залози;
- проаналізувати сучасні алгоритми розпізнавання зображень;
- дослідити програмні засоби розпізнавання біомедичних зображень на основі нейронних мереж;
- дослідити топології згорткових нейронних мереж;
- проаналізувати алгоритми розпізнавання зображень в згорткових нейронних мережах;
- дослідити процес навчання згорткових нейронних мереж;
- розробити програмний засіб класифікації біомедичних зображень передракових станів молочної залози на основі згорткових нейронних мереж;
- виконати тестування розробленого модуля.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

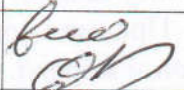

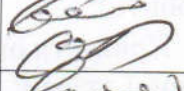
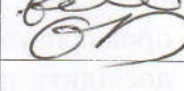
- актуальність роботи, наукова новизна;
- мета роботи, об'єкт та предмет дослідження, практичне значення;
- гістологічні зображення раку молочної залози та їх класифікація;
- характеристики гістологічних зображень;
- порівняння програмних засобів розпізнавання зображень;
- штучні нейронні мережі;
- архітектура згорткових нейронних мереж;
- принципи функціонування згорткових нейронних мереж;
- навчання згорткових нейронних мереж;
- проектування згорткової нейронної мережі;
- структура розробленого програмного модуля;
- результати досліджень.

7. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		Завдання видав	Завдання прийняв
Антиплагіат	Мельник Г.М., доцент		
Нормо-контроль	Гураль І. В., викладач		

7. Дата видачі завдання « 13 » лютого 2016 р.

КАЛЕНДАРНИЙ ПЛАН

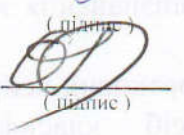
№ з/п	Назва етапів дипломної роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз алгоритмів і програмних засобів розпізнавання зображень	3.11.2016 – 1.01.2017	
2	Алгоритми розпізнавання зображень на основі згорткових нейронних мереж	2.01.2017 – 31.05.2017	
3	Програмна реалізація алгоритмів розпізнавання гістологічних зображень	1.06.2017 – 25.01.2018	
4	Нормоконтроль, попередній захист	16.01.2018 – 2.02.2018	
5	Захист	5.02.2018	

Завдання прийняв до виконання



Боднар А.Р.

Керівник магістерської роботи


(підпис)

д.т.н., проф. Березький О.М.

ЗМІСТ

Вступ.....	6
1 Аналіз алгоритмів і програмних засобів розпізнавання зображень.....	8
1.1 Гістологічні зображення та їх характеристики.....	8
1.2 Аналіз алгоритмів розпізнавання зображень.....	13
1.3 Штучні нейронні мережі.....	22
1.4 Програмні засоби розпізнавання зображень на основі нейронних мереж.....	29
1.5 Постановка завдання.....	34
2 Алгоритми розпізнавання зображень на основі згорткових нейронних мереж.....	35
2.1 Топології згорткових нейронних мереж.....	35
2.2 Алгоритми розпізнавання зображень в згорткових нейронних мережах.....	42
2.3 Алгоритми навчання згорткових нейронних мереж.....	49
3 Програмна реалізація алгоритмів розпізнавання гістологічних зображень.....	56
3.1 Функційна структура програмного модуля.....	56
3.2 UML-діаграми та опис реалізації.....	59
3.3 Програмний модуль розпізнавання гістологічних зображень.....	64
3.4 Тестова вибірка та тестування програмного модуля.....	74
Висновки.....	80
Список використаних джерел.....	81
Додаток А – Діаграма класів.....	82

Додаток Б -	83
Додаток В – Вихідний код.....	84

ВСТУП

Актуальність роботи. За уточненими даними Національного канцер-реєстру (НКР) у 2015 р. було зареєстровано 135 307 нових випадків захворювання на злоякісні новоутворення (ЗН); загальний показник захворюваності на ЗН склав 372,4 на 100 тис. населення, тобто в порівнянні з 2014 р. зменшився у жіночого населення – на 1,7 %, у чоловічого – на 4,7 % [1]. Приріст захворюваності за грубим показником зареєстровано лише в 4-х областях – Київська, Львівська, Сумська та Чернігівська. Встановлено також, що в 13-ти областях та м. Київ показник захворюваності був вищим за середньоукраїнське значення. В жіночій популяції найвищий рівень захворюваності було зареєстровано у Кіровоградській, Запорізькій, Сумській областях та м. Київ (409,3- 442,8 на 100 тис. жіночого населення); перевищення середньоукраїнського рівня захворюваності зареєстровано в 11-ти регіонах, співвідношення мінімального значення показника до максимального склало 1,8 [1].

При дослідженні вікової структури захворілого та померлого від ЗН населення України встановлено, що в 2014 р. серед чоловіків 19,6% захворілих та 17,1% померлих від раку були особами працездатного віку, серед жінок – 27,0% і 18,8%, відповідно, тобто, виявлені особливості ураження населення України злоякісними новоутвореннями свідчать про беззаперечний вплив цієї патології на трудовий та репродуктивний потенціал нації.

Методи досліджень. Діагностування ракових захворювань зазвичай вимагає цитологічного та гістологічного дослідження. Цитологічний метод діагностики захворювань визначив себе як безпечний, простий, швидкий, не дорогий та достатньо інформативний, що дозволяє використовувати його для морфологічного встановлення діагнозу при діагностуванні захворювань молочної залози, органів грудної порожнини, лімфатичних вузлів, м'яких тканин. Цей метод дозволяє встановити характер ураження, гістогенез. Інформативність отриманого матеріалу дозволяє встановити діагноз, від'ємні результати не виключають

наявність злякисного процесу. Повторне отримання матеріалу дозволяє з більшим ступенем достовірності встановити діагноз. Проведення гісто- та цитологічних перевірок - найбільш ефективний метод контролю якості цитологічних досліджень [2].

У випадках відсутності цитологічного підтвердження діагнозу раку показана біопсія - секторальна резекція молочної залози з гістологічним дослідженням препарату.

Мета роботи. Метою роботи є розроблення архітектури згорткової нейронної мережі, яка повинна здійснювати класифікацію гістологічних зображень патологічних процесів молочної залози з високою точністю.

Постановка задачі. Для досягнення поставленої мети необхідно розв'язати такі задачі:

- Здійснити вибірку тестових гістологічних зображень;
- розробити структуру згорткової нейронної мережі;
- провести навчання згорткової нейронної мережі, використовуючи тестову вибірку з прокласифікованими зображеннями;
- провести тестування на контрольній вибірці і проаналізувати його результати.

Об'єкт дослідження. Гістологічні зображення раку молочної залози.

Предмет дослідження. Алгоритми класифікації гістологічних зображень, згорткові нейронні мережі.

Наукова новизна. Розроблено архітектуру згорткової нейронної мережі, яка здійснює класифікацію гістологічних зображень раку молочної залози з точністю 84%.

Практичне значення. Здійснено програмну реалізацію модуля класифікації гістологічних зображень раку молочної залози на основі згорткових нейронних мереж засобами мови програмування Java та бібліотеки Deeplearning4J, а також здійснено тестування на основі бази даних гістологічних зображень.

1 АНАЛІЗ АЛГОРИТМІВ І ПРОГРАМНИХ ЗАСОБІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

1.1 Гістологічні зображення та їх характеристики

Термін «гістологія» був запропонований німецьким вченим Р.Майєром у 1819 році. Під цим терміном розумілася наука про тканини багатоклітинних тварин та людини [3].

Тканиною називають групу клітин, схожих формою, розмірам і функціям і по продуктах своєї життєдіяльності. У всіх рослин і тварин, за винятком найпримітивніших, тіло складається з тканин, причому у вищих рослин і у високоорганізованих тварин тканини відрізняються великою різноманітністю структури і складністю своїх продуктів; поєднуючись один з одним, різні тканини утворюють окремі органи тіла. Гістологію інколи називають мікроскопічною анатомією, оскільки вона вивчає будову (морфологію) організму на мікроскопічному рівні (об'єктом гістологічного дослідження служать дуже тонкі тканинні зрізи і окремі клітини).

Наука гістологія вивчає не тільки тканини, але й клітини, з яких вони складаються, будову органів і систем організму в цілому. Згідно з цим існують такі розділи гістології: цитологія (наука про клітину), загальна гістологія (вивчає тканини), спеціальна гістологія (вивчає будову органів і їх систем).

Предмет гістології людини охоплює вивчення тонкої (мікроскопічної) та ультратонкої (субмікроскопічної) будови структур людського організму, їх розвитку і змін у різноманітних умовах життєдіяльності [3]. Гістологія характеризує структурну організацію макромолекул у ті чи інші мікрооб'єкти.

Гістологічне зображення – це зображення гістологічного препарату, яке зроблене за допомогою цифрового мікроскопу.

До складу об'єкту дослідження (гістологічні зображення раку молочної залози) входять зображення таких видів раку (класів): непроліферативна мастопатія (рисунок 1.1), проліферативна мастопатія (рисунок 1.2), фіброаденома

(рисунок 1.3), внутрішньопротоковий рак (рисунок 1.4), інвазивний рак (1.5).
Нижче наведений якісний опис кожного з цих класів.

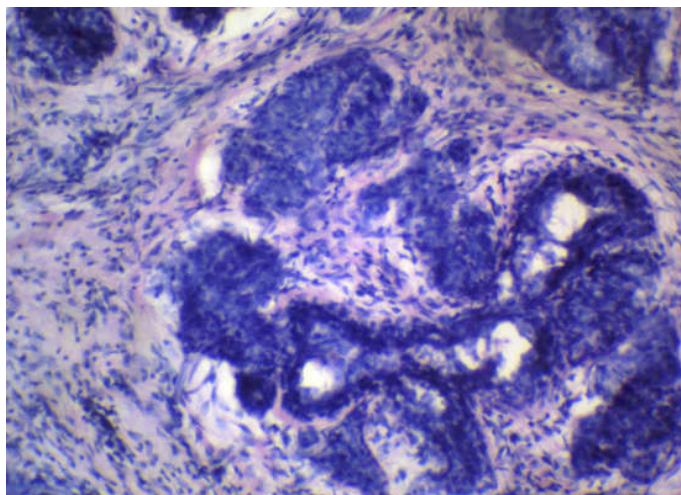


Рисунок 1.1 – Непроліферативна мастопатія раку молочної залози. Гістологія

Мастопатія з переважанням кістозного компонента проявляється кістами, чітко відмежовані від навколишніх тканин залози, утвореними з атрофованих часточок і розширених проток залоз з фіброзними змінами інтерстиціальної тканини.

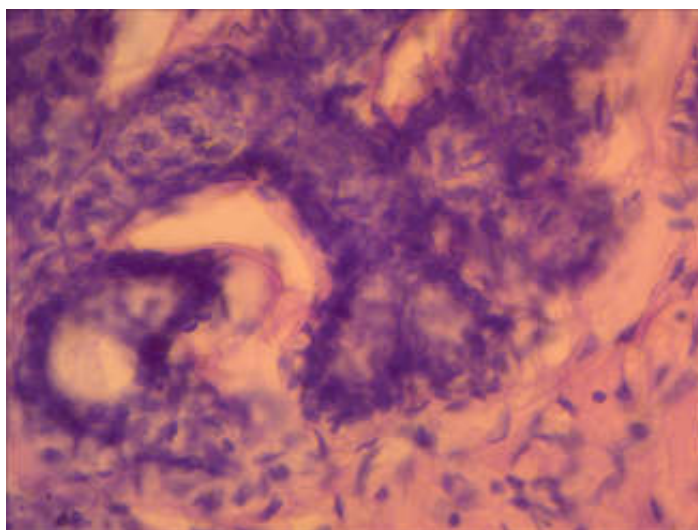


Рисунок 1.2 – Проліферативна мастопатія

Міоепітеліальні клітини зазнають змін. В залежності від свого функціонального стану вони поодинокі або багаточисельні, темні і видовжені, світлі і які містять міхурцеві включення.

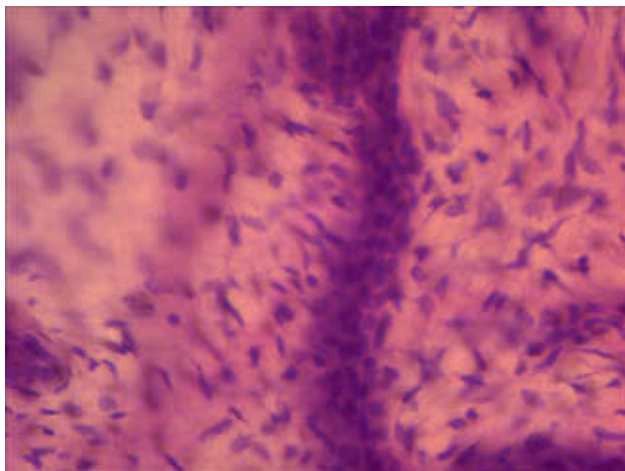


Рисунок 1.3 – Фіброаденома

Фіброаденома має вид інкапсульованого вузла щільної консистенції, волокнистої будови. Мікроскопічно виявляється проліферація альвеол і внутрішньодолькових протоків з розростанням внутрішньодолької сполучної тканини.

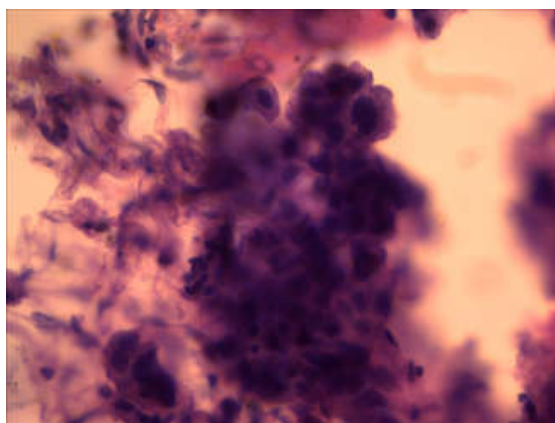


Рисунок 1.4 - Внутрішньопротоковий рак

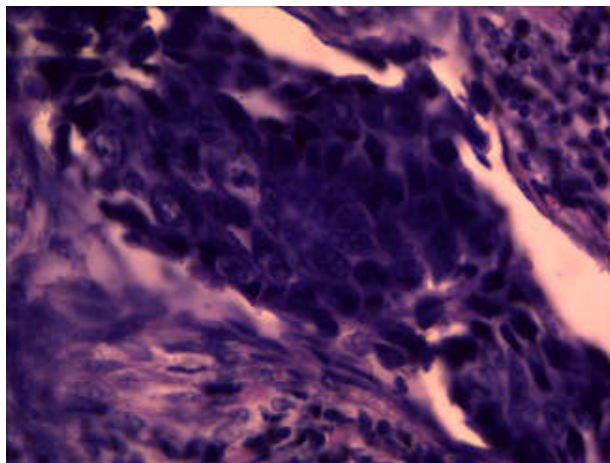


Рисунок 1.5 - Інвазивний рак

Крім загальної гістології існує така наука як кількісна гістологія. Кількісна гістологія - це наука, що вивчає закономірності розвитку та функціонування тканин, використовуючи при цьому кількісні параметри та строгі методи перевірки гіпотез [4]. Кількісну гістологію більш коректно розглядати не як окрему наукову дисципліну, а як перехідний стан власне гістології, на шляху її розвитку від описової до точної науки.

Впровадження елементів кількісного аналізу в практику гістологічного дослідження робить більш ефективним пошук залежностей між структурою і функцією, підвищує точність одержуваних оцінок, знижує вплив суб'єктивного фактора на результати аналізу гістологічного зображення, а також допомагає автоматизувати процедури дослідження і діагностики [5].

Якщо головним інструментом гістології є мікроскоп, то у кількісній гістології цю роль відіграє система аналізу гістологічних зображень, яку спрощено можна уявити собі як мікроскоп, з'єднаний з комп'ютером. Програмна частина таких систем включає десятки алгоритмів, націлених на аналіз різноманітних параметрів тканин та/або клітин. Деякі системи аналізу зображень навіть спроможні приймати діагностичні рішення, що відносить їх до категорії так званих експертних систем [6].

Гістологічні зображення отримуються різними методами: ультрафіолетова мікроскопія, світлооптичний метод, люмінесцентна мікроскопія, фазово-контрастна мікроскопія, поляризаційна мікроскопія та ін. [7]. Однак найбільшого

поширення набув метод світлової мікроскопії, який базується на використанні світлового мікроскопа.

Сучасні застосування кількісної гістології спрямовані на розробку нових методів, що дозволяють позбутися високої варіабельності оцінок, отриманих різними дослідниками; знаходження нових критеріїв оцінки функції тканин, а також маркерів патологічних процесів [8]; розробку алгоритмів для комп'ютеризованих систем, спроможних допомогти лікарю у постановці діагнозу(програмна частина таких систем може включати елементи штучного інтелекту) та ін.

В гістологічному дослідженні кількісна оцінка мікроструктур є необхідною умовою отримання об'єктивних даних про їх стан. Головними кількісними показниками мікроструктур є морфометричні (число структур і їх геометричні параметри) і денситометричні, що відображають концентрацію (оптичну щільність) хімічних речовин в мікроструктурах [9]. Для визначення цих параметрів застосовують морфометричні і спектрофотометричні методи. Морфометрія включає сукупність прийомів і методів визначення геометричних характеристик дослідних об'єктів – гістологічних препаратів (зрізів, мазків, відбитків і т.п.) і мікрофотографій. Вимірювання числа структур, їх площин, діаметрів, периметрів і інших показників здійснюють, наприклад, за допомогою спеціальних морфометричних сіток або окуляр-мікрометра, які вставляють в окуляр світлового мікроскопа [9].

Отже, серед кількісних характеристик гістологічного зображення виділяють:

- Число структур;
- кількість об'єктів;
- геометричні параметри структур;
- розмір структур;
- форма структур;
- розподіл структур та об'єктів у просторі (топографія);
- орієнтація у просторі;

- приналежність до групи (кластерний аналіз);
- топологічні властивості.

Гістологічне дослідження – ведучий, вирішальний і завершальний етап діагностики онкологічних захворювань [10]. Його роль безперервно зростає у зв'язку з прагненням до ранньої діагностики пухлинного процесу і виявленню передпухлинних змін, фонових станів і ранніх стадій розвитку злоякісних пухлин. Вдосконалення ендоскопічної і оперативної техніки, прагнення до виконання органозберігаючих оперативних втручань обумовлюють необхідність ширшого використання морфологічної діагностики пухлин і принципово нових підходів до її проведення.

Гістологічне дослідження включає три основні етапи. Перший і найважливіший етап — встановлення морфологічного діагнозу перед початком лікування (хірургічне, хіміотерапія, променева терапія, гормональна дія). Не менш важлива роль морфолога на другому етапі — етапі проведення термінового гістологічного дослідження під час операції [10]. Нарешті, найбільшу інформацію про характер патологічного процесу можна отримати на завершальному етапі — при плановому морфологічному дослідженні операційного матеріалу.

1.2. Аналіз алгоритмів розпізнавання зображень

Теорія розпізнавання зображено — розділ кібернетики, що розвиває теоретичні основи й методи класифікації і ідентифікації предметів, явищ, процесів, сигналів, ситуацій і т. п. об'єктів, які характеризуються кінцевим набором деяких властивостей і ознак [11].

Необхідність в розпізнаванні зображень виникає в найрізноманітніших областях - від медицини, військової справи і систем безпеки до оцифрування аналогових сигналів.

Розпізнавання образів - це віднесення вихідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних.

При постановці задач розпізнавання намагаються користуватися математичною мовою, прагнучи замінити експеримент логічними міркуваннями і математичними доказами [12].

Класична постановка задачі розпізнавання образів [13]: Дано множину об'єктів. Щодо них необхідно провести класифікацію. Множину представлено підмножинами, які називаються класами. Досліднику відома інформація про класи, опис усієї множини і опис інформації про об'єкт, належність якого до певного класу невідома. Потрібно за наявною інформацією про класи і описі об'єкта встановити - до якого класу належить цей об'єкт.

Тобто, задачею класифікації зображення є побудова функції $y = f(x)$, яка для кожного вектора x ознак дає відповідь y , що означає приналежність зображення до конкретного класу. В даному випадку – функція $f()$ називається класифікатором.

Будь-який класифікатор поділяє простір ознак на регіони R , розділені границями (рисунок 1.6).

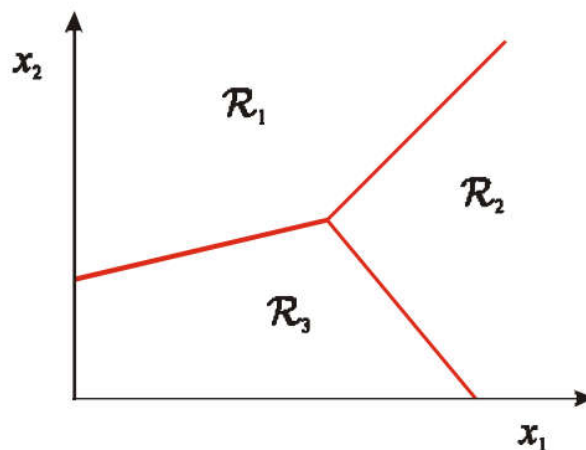


Рисунок 1.6 – Принцип роботи класифікатора

Якщо класів всього два, то завдання називається бінарною класифікацією. Якщо класів кілька - багатокласова (мультикласова) класифікація. Також можуть бути присутніми зразки кожного класу - об'єкти, про які заздалегідь відомо до якого класу вони належать. Такі завдання називають навчанням з учителем, а відомі дані називаються навчальною вибіркою. Якщо класи з самого початку не задані, то перед нами стоїть завдання кластеризації [15].

Існує багато методів та алгоритмів розпізнавання зображень. Найпоширеніші серед них: лінійна класифікація, метод опорних векторів (Support Vector Machine – SVM), нелінійний SVM, AdaBoost, K-Means, Random Forest.

Лінійний класифікатор - алгоритм класифікації, заснований на побудові лінійної розділювальної поверхні. У разі двох класів розділяючою поверхнею є гіперплощина, яка ділить простір ознак на два напівпростори. У разі більшої кількості класів розділяюча поверхня є кусково-лінійна [14].

Застосування лінійного класифікатора зводиться до пошуку лінійної функції (гіперплощини) і поділу додатніх $\{y = +1\}$ та від'ємних $\{y = -1\}$ прикладів (x_i) серед вхідної вибірки (рисунок 1.7).

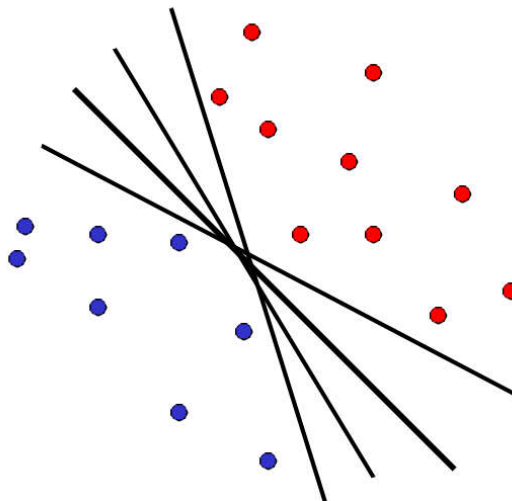


Рисунок 1.7 – Побудова лінійного класифікатора для двох класів

x_i додатні: $x_i \cdot w + b \geq 0$ і x_i від'ємні: $x_i \cdot w + b < 0$.

Навчання лінійного класифікатора може відбуватися за допомогою наступних методів: лінійний дискримінант Фішера, алгоритм перцептрона, логістична регресія, пробіт-апроксимація, експоненціальна апроксимація [14].

Метод опорних векторів — це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням, які називаються опорно-векторними машинами (ОВМ, англ. support vector machines, SVM). Для заданого набору тренувальних зразків, кожен із яких відмічено як належний до однієї чи іншої з двох категорій, алгоритм тренування ОВМ будує модель, яка відносить нові зразки до однієї чи іншої категорії, роблячи це наймовірнішим бінарним лінійним класифікатором [16]. Модель ОВМ є представленням зразків як точок у просторі, відображених таким чином, що зразки з окремих категорій розділено чистою прогалиною, яка є щонайширшою. Нові зразки тоді відображуються до цього ж простору, й робиться передбачення про їхню належність до категорії на основі того, на який бік прогалини вони потрапляють.

Суть методу опорних векторів, як і у випадку лінійного класифікатора, зводиться до пошуку гіперплощини, яка повинна розділити додатні та від'ємні приклади. Різниця лише в тому, що в даному випадку шукана гіперплощина повинна максимізувати відступ між прикладами (рисунк 1.8).

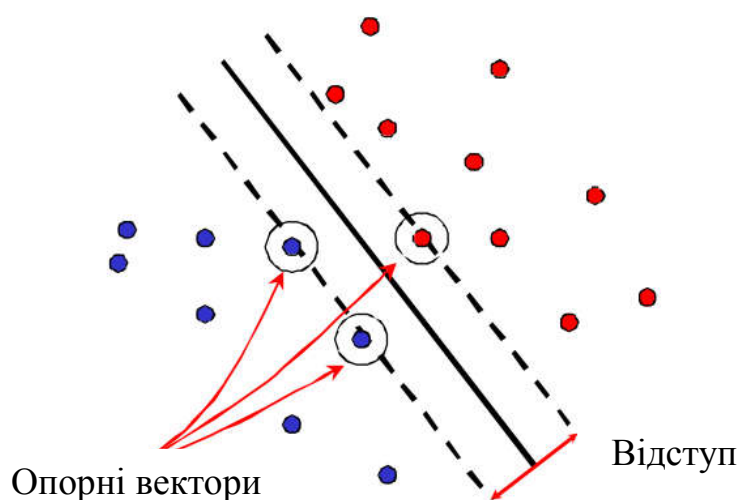


Рисунок 1.8 – Принцип роботи методу опорних векторів

Нехай існує навчальна вибірка:

$$(x_1, y_1), \dots, (x_m, y_m), \quad x_i \in \mathbb{R}^n, \quad y_i \in \{-1, 1\}. \quad (1.1)$$

Метод опорних векторів будує класифікуючу функцію F у вигляді:

$$F(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b), \quad (1.2)$$

де $\langle \cdot, \cdot \rangle$ - скалярний добуток,

\mathbf{w} - вектор нормалі до розділяючої гіперплощини,

b - допоміжний параметр.

Ті об'єкти, для яких $F(\mathbf{x}) = 1$, потрапляють в один клас, а об'єкти з $F(\mathbf{x}) = -1$ - в інший. Вибір саме такої функції не випадковий: будь-яка гіперплощина може бути задана у вигляді $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ для деяких \mathbf{w} і b .

Далі потрібно вибрати такі \mathbf{w} і b які максимізують відстань до кожного класу. Можна підрахувати, що дана відстань дорівнює $\frac{1}{\|\mathbf{w}\|}$. Проблема знаходження максимуму $\frac{1}{\|\mathbf{w}\|}$ еквівалентна проблемі знаходження мінімуму $\|\mathbf{w}\|^2$ (рисунок 1.9).

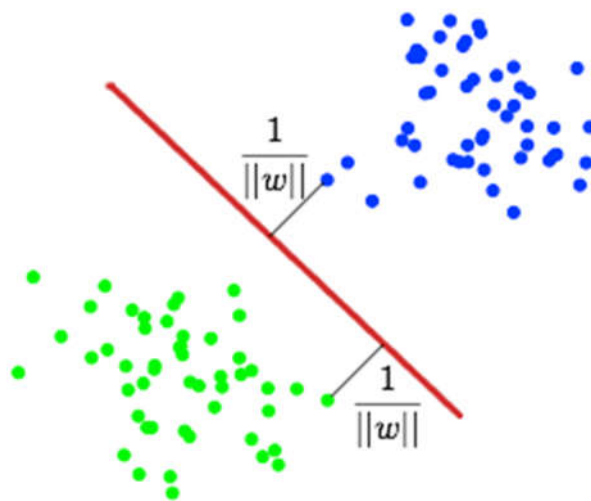


Рисунок 1.9 – Максимізація відступу

Запишемо все це у вигляді завдання оптимізації:

$$\begin{cases} \arg \min_{\mathbf{w}, b} \|\mathbf{w}\|^2, \\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \quad i = 1, \dots, m. \end{cases}, \quad (1.3)$$

яке є стандартною задачею квадратичного програмування і вирішується за допомогою множників Лагранжа.

Нелінійний SVM – частковий випадок, в якому класи не можуть бути лінійно розділені (рисунок 1.10).

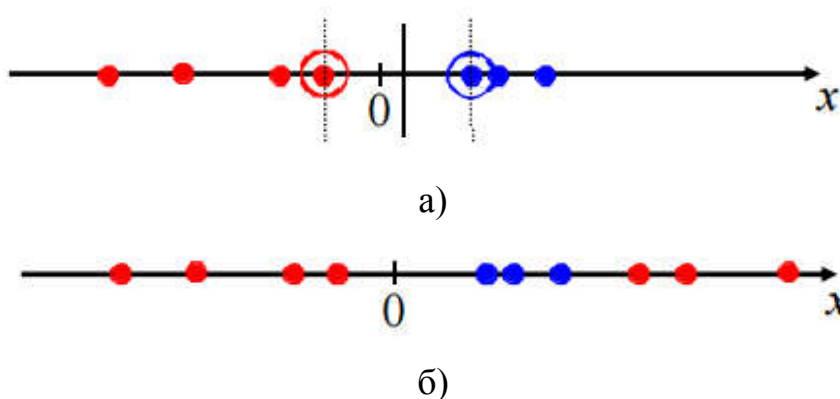


Рисунок 1.10 – Приклад лінійної роздільності (а) та нероздільності (б) даних

В такому випадку можна відобразити дані в просторі більшої розмірності і розділити їх лінійно там або здійснити відображення початкового простору параметрів на деякий багатовимірний простір ознак, де вхідна вибірка лінійно роздільна (рисунок 1.11).

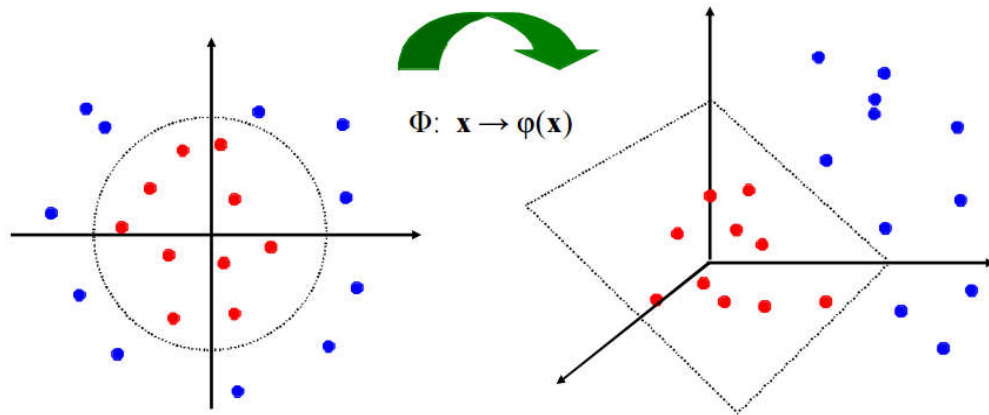


Рисунок 1.11 – Приклад розділення нелінійних даних

Алгоритм AdaBoost (скор. Від adaptive boosting) - алгоритм машинного навчання, запропонований Йоав Фройнд (Yoav Freund) і Робертом Шапіро (Robert Schapire) [17]. Є мета-алгоритмом, в процесі навчання будує композицію з базових алгоритмів навчання для поліпшення їх ефективності. AdaBoost є алгоритмом адаптивного бустінгу в тому сенсі, що кожен наступний класифікатор будується по об'єктах, які погано класифікуються попередніми класифікаторами.

AdaBoost викликає слабкий класифікатор в циклі. Після кожного виклику оновлюється розподіл ваг, які відповідають важливості кожного з об'єктів навчальної вибірки для класифікації [17]. На кожній ітерації ваги кожного невірно класифікованого об'єкта зростають, таким чином новий класифікатор «фокусує свою увагу» на цих об'єктах.

k-means (метод k-середніх) - найбільш популярний метод кластеризації та класифікації. Був винайдений в 1950-х роках математиком Гуго Штейнгаузом і майже одночасно Стюартом Ллойдом.

Алгоритм розбиває множину елементів векторного простору на заздалегідь відоме число кластерів k.

Основна ідея полягає в тому, що на кожній ітерації переобчислюють центр мас для кожного кластера, отриманого на попередньому кроці, потім вектори розбиваються на кластери знову відповідно до того, який з нових центрів виявився ближчим за обраною метриці [18].

Алгоритм завершується, коли на якийсь ітерації не відбувається зміни центру мас кластерів. Це відбувається за кінцеве число ітерацій, так як кількість можливих розбиттів кінцевого безлічі звичайно, а на кожному кроці сумарна квадратичне відхилення V не збільшується, тому зациклення неможливо.

Принцип алгоритму полягає в пошуку таких центрів кластерів та наборів елементів кожного кластера при наявності деякої функції $\Phi(\circ)$, що виражає якість поточного розбиття множини на k кластерів, коли сумарне квадратичне відхилення елементів кластерів від центрів цих кластерів буде найменшим:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2, \quad (1.4)$$

де k – число кластерів,

S_i – отримані кластери,

μ_i – центри мас векторів $x_j \in S_i$ [18].

В початковий момент роботи алгоритму довільним чином обираються центри кластерів, далі для кожного елемента множини ітеративно обраховується відстань від центрів з приєднанням кожного елемента до кластера з найближчим центром [18]. Для кожного з отриманих кластерів обчислюються нові значення центрів, намагаючись при цьому мінімізувати функцію $\Phi(\circ)$, після чого повторюється процедура перерозподілу елементів між кластерами.

Random forest — алгоритм машинного навчання, запропонований Лео Брейманом і Адель Катлер, що полягає у використанні комітету (ансамблю) вирішальних дерев. Алгоритм поєднує в собі дві основні ідеї: метод бегінга Брейман і метод випадкових підпросторів, запропонований Tin Kam Ho. Алгоритм застосовується для задач класифікації, регресії і кластеризації.

Нехай навчальна вибірка складається з N прикладів, розмірність простору ознак дорівнює M , і заданий параметр m .

Усі дерева комітету будуються незалежно один від одного за такою процедурою [19]:

1. Згенеруємо випадкову підвибірку з повторенням розміром n з навчальної вибірки. (Таким чином, деякі приклади потраплять в неї кілька разів, а приблизно $N / 3$ прикладів не ввійдуть у неї взагалі).

2. Побудуємо вирішальне дерево, яке класифікує приклади даної підвибірки, причому в ході створення чергового вузла дерева будемо вибирати ознаку, на основі якої проводиться розбиття, не з усіх M ознак, а лише з m випадково вибраних. Вибір найкращого з цих m ознак може здійснюватися різними способами.

3. Дерево будується до повного вичерпання підвибірки і не піддається процедурі прунінга (англ. pruning — відсікання гілок).

Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету відносить об'єкт, який класифікується до одного з класів, і перемагає клас, за який проголосувало найбільше число дерев [19].

Оптимальне число дерев підбирається таким чином, щоб мінімізувати помилку класифікатора на тестовій вибірці. У разі її відсутності, мінімізується оцінка помилки *out-of-bag*: частка прикладів навчальної вибірки, неправильно класифікованих комітетом, якщо не враховувати голоси дерев на прикладах, що входять в їх власну навчальну підвибірку.

1.3. Штучні нейронні мережі

Штучні нейронні мережі (ШНМ) являють собою модель пристрою паралельних обчислень, який складається із множини взаємодіючих простих процесорів, які мають справу тільки із сигналами, які періодично надходять до них і відсилаються іншим [20].

Розробка штучних нейронних мереж почалася ще на початку ХХ століття, але тільки в 90-х роках вони отримали широке поширення. Створення ШНМ було спричинено спробами зрозуміти принципи роботи людського мозку.

Нейронна мережа являється сукупністю елементів, з'єднаних деяким чином так, щоб між ними забезпечувалась взаємодія. Ці елементи називаються нейронами або вузлами. Вихідний сигнал елементу може відсилатися іншим елементам по зваженим зв'язкам, з кожним з яких пов'язаний ваговий коефіцієнт чи вага [20]. В залежності від значення вагового коефіцієнта сигнал, який передається, або підсилюється, або пригнічується. Елемент нейронної мережі схематично представлений на рисунку 1.12.

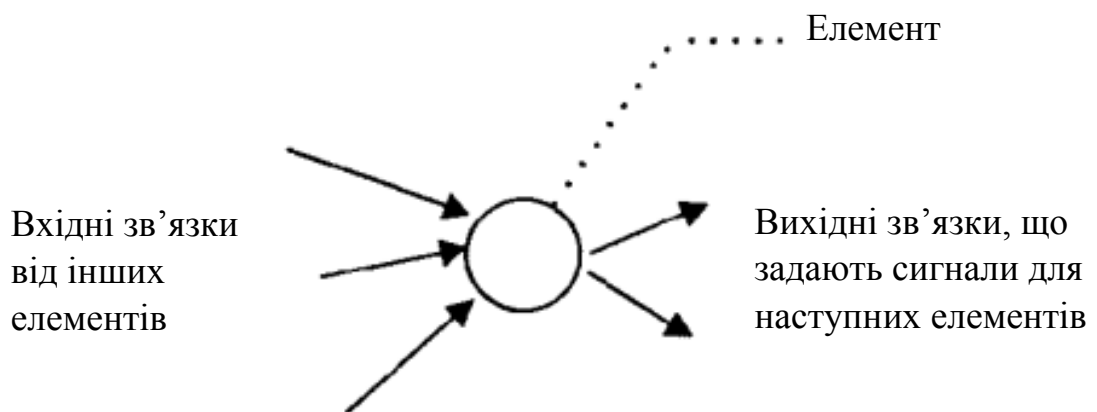


Рисунок 1.12 – Схематичне представлення окремого елемента ШНС

Зазвичай вагові коефіцієнти зв'язків визначають шляхом навчання. Однак, вагові коефіцієнти можна визначити і без проведення навчання нейронної мережі, але якраз основна перевага нейронних мереж заключається в їх здатності

навчатися виконанню певної задачі на основі тих даних, які мережа буде отримувати в процесі реальної роботи.

Використання нейронних мереж забезпечує наступні корисні властивості систем.

1). Нелінійність (nonlinearity). ШНМ можуть бути лінійними та нелінійними. Нейронні мережі, які побудовані із з'єднань нелінійних нейронів, являються нелінійними. Нелінійність є дуже важливою властивістю, особливо якщо сам фізичний механізм, що відповідає за формування вхідного сигналу, також являється нелінійним (наприклад, аудіо-дані).

2). Відображення вхідної інформації у вихідну (input-output mapping). Одною із популярних парадигм навчання є навчання із вчителем (supervised learning). Під цим розуміється зміна синаптичних ваг на основі маркованих навчальних прикладів (training sample). Кожен приклад складається із вхідного сигналу і відповідного йому бажаного відклику (desired response) [21]. З цієї множини випадковим чином вибирається приклад, а нейронна мережа модифікує синаптичні ваги для мінімізації розбіжностей бажаного вихідного сигналу і формованого мережею згідно вибраного статистичного критерію. Це відбувається до тих пір, поки зміни синаптичних ваг не стануть незначними.

3). Адаптивність (adaptivity). Нейронні мережі володіють здатністю адаптувати свої синаптичні ваги до змін оточуючого середовища. Чим вищі адаптивні властивості системи, тим стійкішою буде її робота в нестационарному середовищі.

4). Очевидність відповіді (evidential response). В контексті задачі класифікації зображень можна розробити нейронну мережу, яка збирає інформацію не тільки для визначення конкретного класу, але і для збільшення достовірності прийнятого рішення. В результаті ця інформація може використовуватися для виключення сумнівних рішень, що підвищить продуктивність нейронної мережі [21].

5). Контекстна інформація (contextual information). Знання представляються в самій структурі нейронної мережі за допомогою її стану активації. Кожен

нейрон мережі потенційно може бути схильний до впливу всіх інших її нейронів. Як наслідок, існування нейронної мережі безпосередньо пов'язане з контекстною інформацією.

6). Відмовостійкість (fault tolerance). При несприятливих умовах продуктивність нейронних мереж падає незначно. Враховуючи розподілений характер зберігання інформації в нейронній мережі, тільки серйозні пошкодження структури ШНМ можуть значно вплинути на її роботу [24].

7). Масштабованість (VLSI Implementability). Паралельна структура нейронних мереж потенційно прискорює вирішення деяких задач і забезпечує масштабованість нейронних мереж в рамках технології VLSI (very-large-scale-integrated). Одною з переваг цієї технології є можливість представити достатньо складну поведінку з допомогою ієрархічної структури [25].

8). Одноманітність аналізу і проектування (Uniformity of analysis and design). Нейронні мережі являються універсальним механізмом обробки інформації. Це означає, що одне і те ж рішення нейронної мережі може використовуватися в багатьох предметних областях.

9). Аналогія з нейробиологією (Neurobiological analogy). Будова нейронних мереж визначається аналогією з людським мозком. В свою чергу, нейробиологи розглядають ШНМ як засіб моделювання фізичних явищ.

Якість роботи нейронної мережі сильно залежить від наданого їй в процесі навчання набору навчальних даних. Навчальні дані мають бути типовими для задачі, вирішенню якої навчається мережа [20].

Історично першою публікацією, що заклала підвалини для створення штучних нейронів та нейронних мереж, вважають роботу Уоррена С. МакКаллока та Вальтера Піттса [26]. У цій роботі було започатковано теорію, в основі якої лежав той факт, що всі аспекти нервової діяльності можна моделювати за допомогою мережі елементів, які мають два стійкі стани. Модель такого елемента, яка показана на рисунку 1.13, отримала назву формального нейрона.

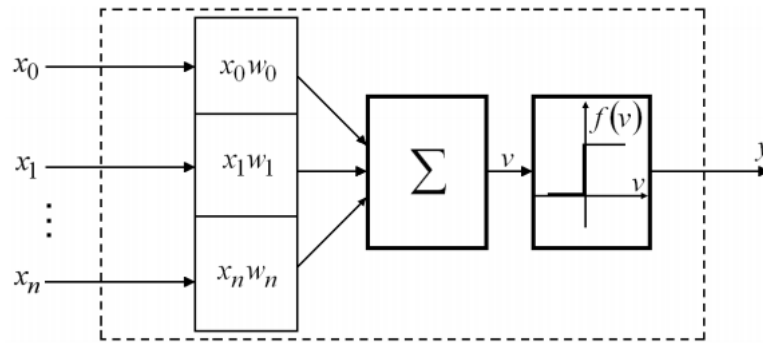


Рисунок 1.13 – Формальний нейрон

В нейробиології електричний синапс — це ділянка з'єднання двох нейронів, яка характеризується високою провідністю. В ШНМ синапсом називають зв'язок двох нейронів, який характеризується певним ваговим коефіцієнтом [23].

Моделювання функцій синапсів відбувається шляхом масштабування вхідних сигналів (x_0, x_1, \dots, x_n) за допомогою вагових коефіцієнтів (w_0, w_1, \dots, w_n) . Одержані сигнали подаються на вхід суматора, який виконує їх обробку за формулою:

$$v = \sum_{i=0}^n w_i x_i. \quad (1.5)$$

Змінна v є аргументом порогової активаційної функції:

$$y = f(v). \quad (1.6)$$

Активаційна функція, що вперше була запропонована в [27], має вигляд:

$$y = \begin{cases} 1 & \text{при } v \leq 0, \\ 0 & \text{при } v > 0. \end{cases} \quad (1.7)$$

У сучасній літературі зустрічається велика кількість парадигм штучних нейронних мереж, елементи яких реалізують різні активаційні функції. Найбільш

поширеною є сигмоїдальна функція, що може бути представлена в дискретному та аналоговому варіанті. Прикладом найпростішої активаційної функції є дискретний сигмоїд із параметром, що має назву функції Хевісайда або тета-функції:

$$y = \begin{cases} 1 & \text{при } v \leq a \\ 0 & \text{при } v > a. \end{cases} \quad (1.8)$$

Якщо величина збудження v нейрона з активаційною функцією Хевісайда не перевищує значення параметра a , то нейрон залишається пасивним, а при перевищенні порога видає фіксоване значення функції, що приймається за логічну одиницю [23].

У випадку застосування порогових активаційних функцій вихідні сигнали ШНМ мають дискретний характер, а задачі, що при цьому можуть бути розв'язні, називають задачами класифікації. При застосуванні формальних нейронів із сигмоїдальною або іншою неперервною активаційною функцією на ШНМ доцільно розв'язувати задачі регресії [22]. Такий поділ є до деякої міри умовним, оскільки задачі класифікації можуть використовувати також неперервні функції активації, значення яких трактуються як імовірності приналежності до відповідних класів.

Архітектури сучасних нейронних мереж найчастіше поділяють на три категорії:

- мережі з повним набором міжнейронних зв'язків;
- мережі з фіксованим індексом оточення;
- мережі з пошаровою структурою

У ШНМ із повним набором міжнейронних зв'язків забезпечується можливість взаємодії кожного нейрона мережі з будь-яким іншим. На рисунку 1.14 представлено приклад повного з'єднання чотирьох нейронів.

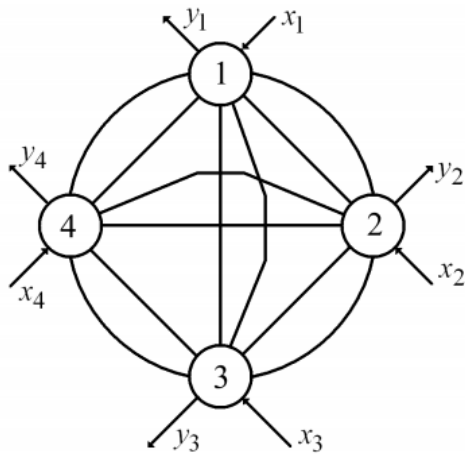


Рисунок 1.14 – Повне з'єднання нейронів

Структури з повним з'єднанням є узагальненими структурами, оскільки всі інші довільні об'єднання нейронів можуть розглядатися як підмножини даної структури [23]. Тому ШНМ із повним з'єднанням є універсальним середовищем для реалізації мережних алгоритмів. У випадку, коли необхідно використовувати структури з великою кількістю нейронів, застосовують кліткові структури з фіксованим індексом оточення. На рисунку 1.15 наведено приклад структури такого типу з індексом оточення 4.

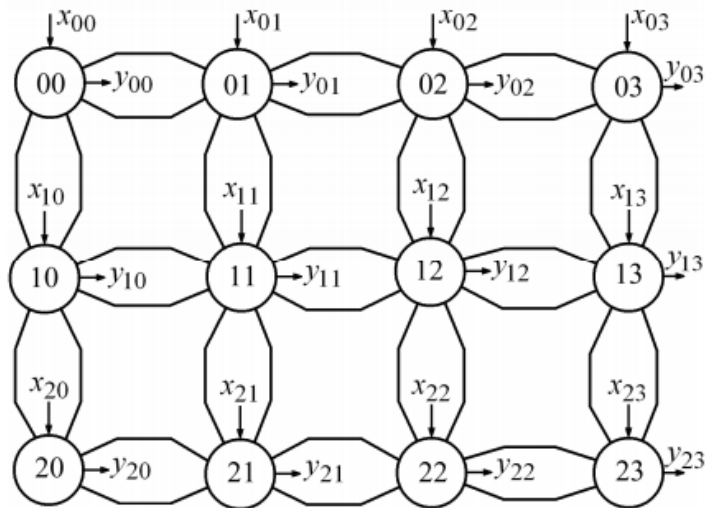


Рисунок 1.15 – Кліткова нейронна мережа з індексом оточення 4

Ще однією важливою характеристикою нейронних мереж із постійним індексом оточення є модульність. Потужність структури у цьому випадку може

нарощуватись простим додаванням елементів без зміни ідеології алгоритму, що на ній працює.

Найбільш поширеними є структуровані за шарами нейронні мережі, які, в залежності від свого функціонального призначення, можуть містити однотипні або різнотипні нейрони. Виходячи з шарової структури ШНМ, характер міжнейронних зв'язків має свої міжшарові та внутрішньшарові особливості.

Багатошарові мережі з повним з'єднанням забезпечують можливість передачі інформації з кожного нейрона попереднього шару на будь-який нейрон наступного.

Найчастіше це — багатошарові перцептрони. У випадку прямого міжшарового поширення інформація передається однонаправлено у напрямку зростання номера шару. Таким чином, однонаправленість зв'язків призводить до побудови виключно ієрархічних структур, у яких обробка інформації розподіляється по рівнях. За кожний рівень ієрархічної обробки інформації відповідає свій шар нейронів. Вихідна інформація більш високого рівня попереднього шару є вхідною для нейронів наступного шару, який забезпечує глибший рівень обробки.

На рисунку 1.16 показаний загальний випадок одношарової нейронної мережі, у якій кожен нейрон приймає участь в обробці всіх елементів вхідного вектора даних.

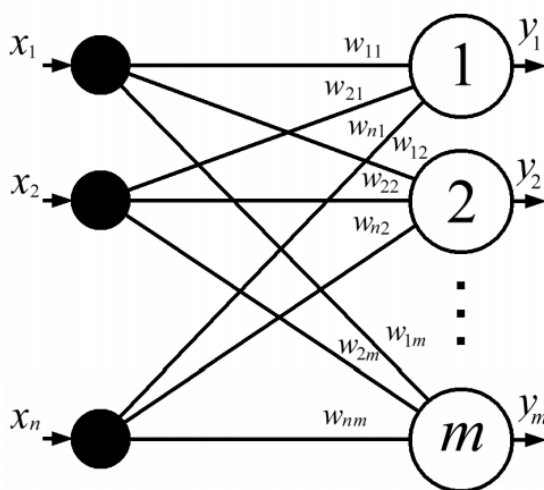


Рисунок 1.16 – Одношарова нейронна мережа

Дана мережа складається з m нейронів, які одночасно здатні прийняти вхідний вектор сигналів $X = (x_1, \dots, x_i, \dots, x_n)$. Для розмноження елементів x_i цього вектора використовують спеціальні пристрої, які показані зліва від нейронів. Ці пристрої не виконують обробки інформації, тому не вважаються шаром нейронної мережі. Згідно з моделлю формального нейрона, кожен з його вхідних сигналів множать на ваговий коефіцієнт w_{ij} , де i — поточний номер елемента вектора X , а j — поточний номер нейрона. Всі вагові коефіцієнти одношарової нейронної мережі утворюють матрицю вагових коефіцієнтів W [23].

Тоді вектор аргументів визначається як добуток $V = X \cdot W$, а вектор вихідних сигналів є вектором значень активаційних функцій ($Y = F(V)$).

Згадані функції можуть бути однаковими для всіх нейронів, що складають мережу. У цьому випадку її називають гомогенною. Мережу, в якій вигляд активаційної функції залежить від номера нейрона, називають гетерогенною [20].

1.4. Програмні засоби розпізнавання зображень на основі нейронних мереж

Нейромережеве програмне забезпечення використовується для моделювання, дослідження, розробки і застосування штучних нейронних мереж, штучного інтелекту і машинного навчання.

Існує багато програмних засобів для розпізнавання зображень на основі нейронних мереж. Одним з найпоширеніших серед них є Matlab.

Matlab — одна з сучасних систем автоматизації математичних розрахунків, яка побудована на розширеному представленні й застосуванні матричних операцій. Це знайшло відображення в назві системи — MATrix LABoratory — матрична лабораторія [28]. Однак синтаксис мови програмування системи продуманий настільки ретельно, що ця орієнтація майже не відчувається тими користувачами, яких не цікавлять безпосередньо матричні обчислення.

Матриці широко застосовуються в складних математичних розрахунках, наприклад, при рішенні задач лінійної алгебри і математичного моделювання статичних і динамічних систем і об'єктів. Вони є основою автоматичного складання й рішення рівнянь стану динамічних об'єктів і систем. Прикладом може служити розширення Matlab — Simulink [28]. Це істотно підвищує інтерес до системи Matlab, що увібрала в себе кращі досягнення в області швидкого рішення матричних задач.

Однак сьогодні Matlab далеко вийшла за межі спеціалізованої матричної системи і стала однією з найбільш могутніх універсальних інтегрованих обчислювальних систем.

В цілому, Matlab — це унікальна колекція реалізацій сучасних чисельних методів комп'ютерної математики, створених за останні три десятки років. Вона увібрала в себе і досвід, правила і методи математичних обчислень, які були накопичені за тисячі років розвитку математики. Це сполучається з могутніми засобами графічної візуалізації і навіть анімаційної графіки. Систему з прикладеною до неї об'ємною документацією цілком можна розглядати як фундаментальний багатотомний електронний довідник по математичному забезпеченню комп'ютерів — від масових персональних комп'ютерів до суперкомп'ютерів [29].

Для роботи з нейронними мережами у Matlab створено Neural Network Toolbox. Neural Network Toolbox забезпечує всебічну підтримку проектування, навчання та моделювання багатьох відомих мережевих парадигм, від базових моделей перцептрона до найсучасніших асоціативних і самоорганізованих мереж [30].

На рисунку 1.17 наведено приклад вікна у Neural Network Toolbox.

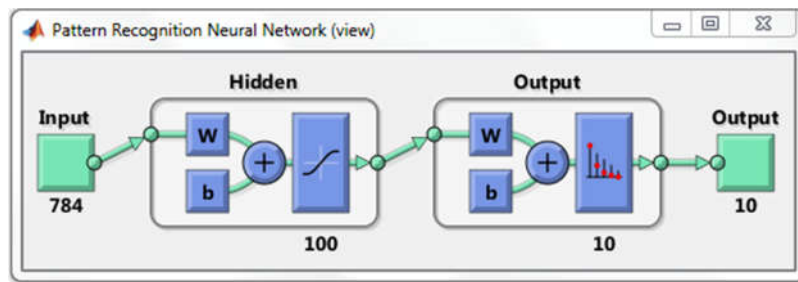


Рисунок 1.17 – Приклад вікна у Neural Network Toolbox

Пакет фірми "The MathWorks" MATLAB також надає користувачам можливість роботи з нейронними мережами. Включений в стандартну комплектацію MATLAB набір інструментів надає широкі можливості для роботи з нейронними мережами всіх типів. Використання "Neural Network Toolbox" спільно з іншими засобами MATLAB відкриває широкий простір для ефективного комплексного використання сучасних математичних методів для вирішення найрізноманітніших завдань прикладного та наукового характеру [31]. В даний час доступна версія 4.0 "Neural Network Toolbox", яку поставляють в складі MATLAB 6.0.

Peltarion Synapse – це компонентно-орієнтоване середовище розробки для нейронних мереж і адаптивних систем. Даний програмний комплекс дозволяє проводити дослідження у таких напрямках як data mining, статистичний аналіз, візуалізація, попереднє оброблення зображень, створення та навчання нейронних мереж та адаптивних систем [32].

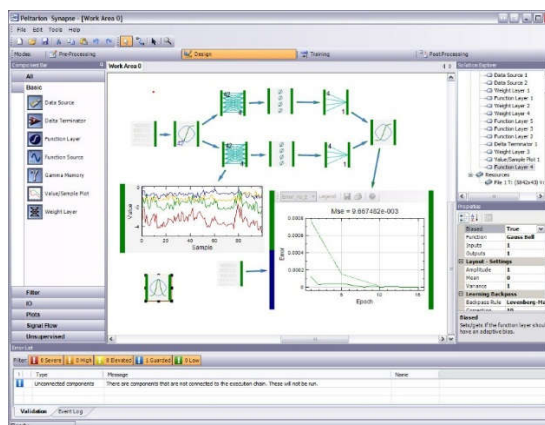


Рисунок 1.18 – Приклад вікна в ПЗ Peltarion Synapse

Цикл розробки в Synapse базується на основі канонічного циклу інтелектуального аналізу даних.

NeuroSolutions – це середовище розробки нейронних мереж, розроблене компанією NeuroDimension. Воно поєднує в собі модульний, компонентно-орієнтований інтерфейс проектування нейронних мереж [33].

На рисунку 1.19 наведено приклад процесу проектування нейронної мережі у середовищі NeuroSolutions.

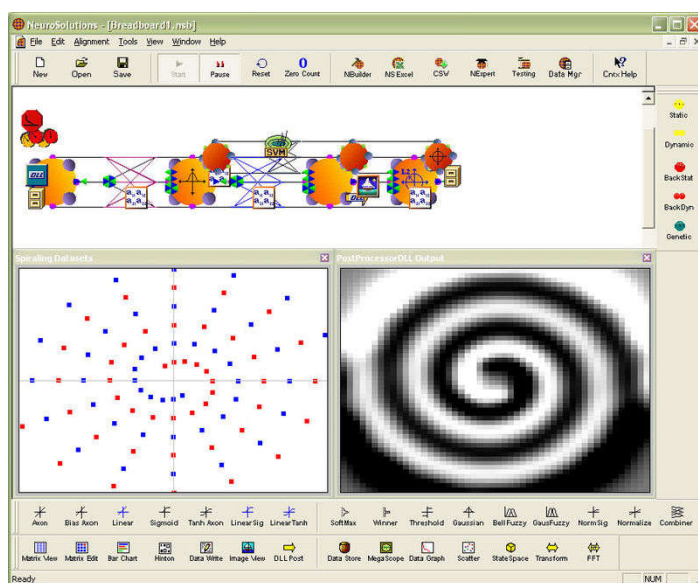


Рисунок 1.19 – Процес проектування нейронної мережі у середовищі NeuroSolutions

Вище наведене програмне забезпечення використовується для проектування, навчання і розгортання моделей нейронних мереж (навчання з вчителем і без вчителя) для виконання широкого спектру завдань, таких як класифікація, інтелектуальний аналіз даних, апроксимація функцій, множинна регресія та прогнозування часових рядів.

Deeplearning4j (DL4J) – це перша у своєму роді бібліотека з відкритим кодом (Open-source), яка використовується для машинного навчання. Дана бібліотека розроблена на мові програмування Java та Scala [35]. Крім цього, Deeplearning4j підтримує обчислення як на центральному процесорі (CPU), так і на графічному адаптері (GPU). Використання графічного адаптера дозволяє

реалізувати розпаралелення виконання обчислень, що значно збільшує швидкодію та зменшує тривалість цих обчислень.

DL4J застосовується для виявлення шахрайства, рекомендаційних систем (CRM, Adtech), регресії і прогностичного аналізу, розпізнавання зображень та образів, пошук голосу, мови в текст (транскрипції), і планово-попереджувального апаратного моніторингу (виявлення аномалій).

Deeplearning4j включає в себе як розподілене, багато-поточе машинне навчання, так і нормальне однопоточе машинне навчання. Навчання проходить в кластері, що означає, що він може обробляти великі обсяги даних максимально швидко. Мережі проходять навчання паралельно з допомогою ітеративного зменшення, і вони в рівній мірі сумісні з Java, Scala і Clojure. Роль Deeplearning4j як модульного компонента у відкритому стеку робить перший каркас глибинного навчання адаптована для архітектури мікро-послуг [34].

У Deeplearning4j включені засоби захисту від зовнішнього втручання і безпеки від злому, що особливо важливо в фінансових задачах, в промислових системах, в електронній комерції та підприємстві, де застосовується розпізнавання аномалій і розпізнавання образів. Deeplearning4j інтегрується з іншими платформами машинного навчання - такими як RapidMiner, Prediction.io і Weka.

1.5. Постановка завдання

Із поставленого завдання розробити програмний модуль класифікації біомедичних зображень передракових станів молочної залози на основі згорткових нейронних мереж впливають наступні підзавдання:

- 1) привести опис і характеристики гістологічних зображень передракових станів молочної залози;
- 2) проаналізувати сучасні алгоритми розпізнавання зображень;
- 3) дослідити програмні засоби розпізнавання біомедичних зображень на основі нейронних мереж;
- 4) дослідити топології згорткових нейронних мереж.
- 5) розробити програмний модуль класифікації біомедичних зображень передракових станів молочної залози на основі згорткових нейронних мереж.
- 6) виконати тестування розробленого модуля.

2 АЛГОРИТМИ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ НА ОСНОВІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

2.1. Топології згорткових нейронних мереж

В машинному навчанні згорткова нейронна мережа (convolutional neural network, CNN) — це такий тип штучної нейронної мережі прямого поширення, в якому схема з'єднання її нейронів заснована на організації зорової кори тварин, окремі нейрони якої впорядковано таким чином, що вони реагують на області, які покривають зорове поле, частково перекриваючись [35]. Згорткові нейронні мережі мають широке застосування в розпізнаванні зображень та відео, рекомендаційних системах [36] та обробці природної мови.

При застосуванні для розпізнавання зображень згорткові нейронні мережі (ЗНМ) складаються з декількох шарів невеликих збірок нейронів, які обробляють частини вхідного зображення, що називаються рецептивними полями. Виходи цих збірок потім укладаються таким чином, щоби їхні вхідні області перекривалися, для отримання кращого представлення первинного зображення; це повторюється для кожного такого шару. Укладання з перекриттям дозволяє ЗНМ бути терпимими до паралельних перенесень вхідного зображення [37].

Згорткові мережі можуть включати шари локальної або глобальної підвибірки, які поєднують виходи кластерів нейронів. [38] Вони також складаються з різних комбінацій згорткових та повноз'єднаних шарів, із застосуванням поточної нелінійності в кінці кожного шару. Для зниження числа вільних параметрів та покращення узагальнення вводиться операція згортки на малих областях входу.

Ідея згорткових нейронних мереж полягає в чергуванні згорткових шарів (C-layers), субдискретизуючих шарів (S-layers) і наявності повнозв'язних (F-layers) шарів на виході. На рисунку 2.1 приведено першу топологію згорткової нейронної мережі, яка була розроблена Яном Лекуном у 1998 році (LeNet) [35].

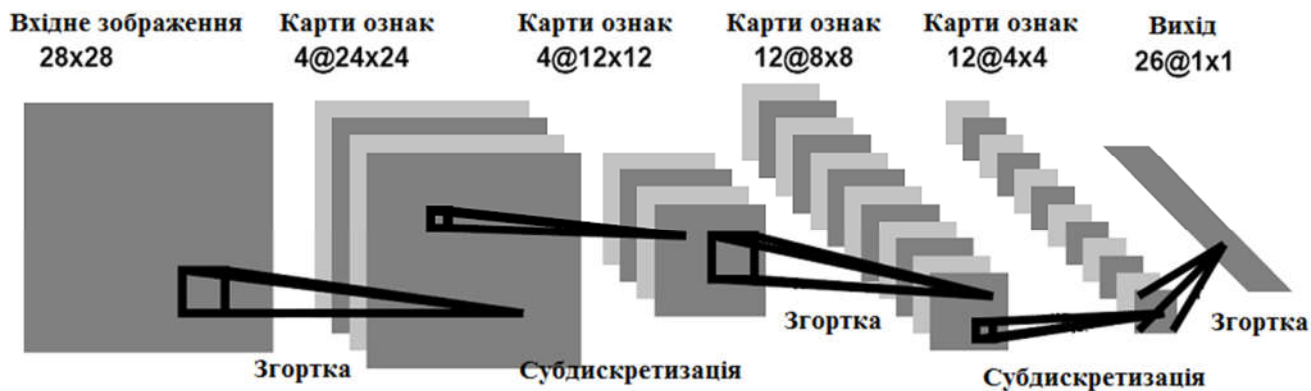


Рисунок 2.1 – Типова архітектура згорткової нейронної мережі

Дана архітектура була розроблена у 2012 році і отримала назву AlexNet. Олексій Крижевський, Ілля Сутскеров та Джефрі Хінтон створили «велику, глибоку згорткову нейронну мережу» [45], яка була використана для виграшу 2012 ILSVRC (Large-Scale VisualNet Challenge). Цей конкурс можна розглядати як щорічну Олімпіаду комп'ютерного зору, де команди з усього світу змагаються, щоб побачити, хто має найкращу модель зору комп'ютера для таких завдань, як класифікація, локалізація, виявлення тощо. 2012 рік ознаменувався першим роком, коли CNN використовувався для досягнення п'яти кращих коефіцієнтів похибки тестування - 15,4%. Помилка 5-го рівня - це швидкість, при якій, з урахуванням зображення, модель не виводить правильну мітку зі своїми п'яти попередніми прогнозами [45]. Наступний кращий запис досяг 26,2% помилки, що стало вражаючим покращенням, яке сильно потрясло общину комп'ютерного бачення. Сейф сказати, що CNN ставали іменами господарів на змаганнях з того часу.

У роботі групи обговорювалася архітектура мережі (яка отримала назву AlexNet). Вони використовували порівняно простий макет, порівняно з сучасними архітектурами. Мережа складалася з 5 згорткових шарів, субдискретизуючих шарів і трьох повнозв'язних шарів [45].

Така архітектура заключає в собі 3 основних парадигми:

- Локальне сприйняття.
- Спільні ваги.

- Субдискретизація [39].

Під локальним сприйняттям розуміється, що на вхід одного нейрона подається не все зображення (або виходи попереднього шару), а лише деяка його область. Такий підхід дозволяє зберігати топологію зображення від шару до шару.

Операція субдискретизація (Subsampling) виконує зменшення розмірності сформованих карт ознак. У даній архітектурі мережі вважається, що інформація про факт наявності шуканої ознаки важливіший точного знання його координат, тому з кількох сусідніх нейронів карти ознак вибирається максимальний і приймається за один нейрон карти ознак зменшеною розмірності. За рахунок цієї операції, крім прискорення подальших обчислень, мережа стає більш інваріантною до масштабу вхідного зображення.

Таким чином, повторюючи один за одним кілька шарів згортки і субдискретизації будується згорткова нейронна мережа. Зазвичай після проходження декількох шарів карта ознак вироджується в вектор або навіть скаляр, але таких карт ознак стає сотні. На виході мережі часто додатково встановлюють кілька шарів повнозв'язної нейронної мережі (перцептрон), на вхід якого подаються кінцеві карти ознак.

Якщо на першому шарі двовимірне ядро згортки проходить по одноканальному зображенню, то на внутрішніх шарах, що складаються з безлічі карт ознак, береться тривимірне ядро розміру $[K \cdot K \cdot N]$, де K - розмір ядра, а N - кількість карт ознак в поточному шарі, формуючи (після проходження функції активації) одну карту ознак наступного шару, що відповідає цьому ядру згортки.

У 2012 році в тій же лабораторії було вдосконалено першу топологію згорткових нейронних мереж. Тепер глибоке навчання включало наступні елементи:

- модель,
- теорія навчання,
- великі дані,
- апаратне забезпечення.

Процес навчання з використанням GPU дозволив значно підвищити кількість параметрів мережі [35]. Отримана мережа отримала назву AlexNet (рисунок 2.2).

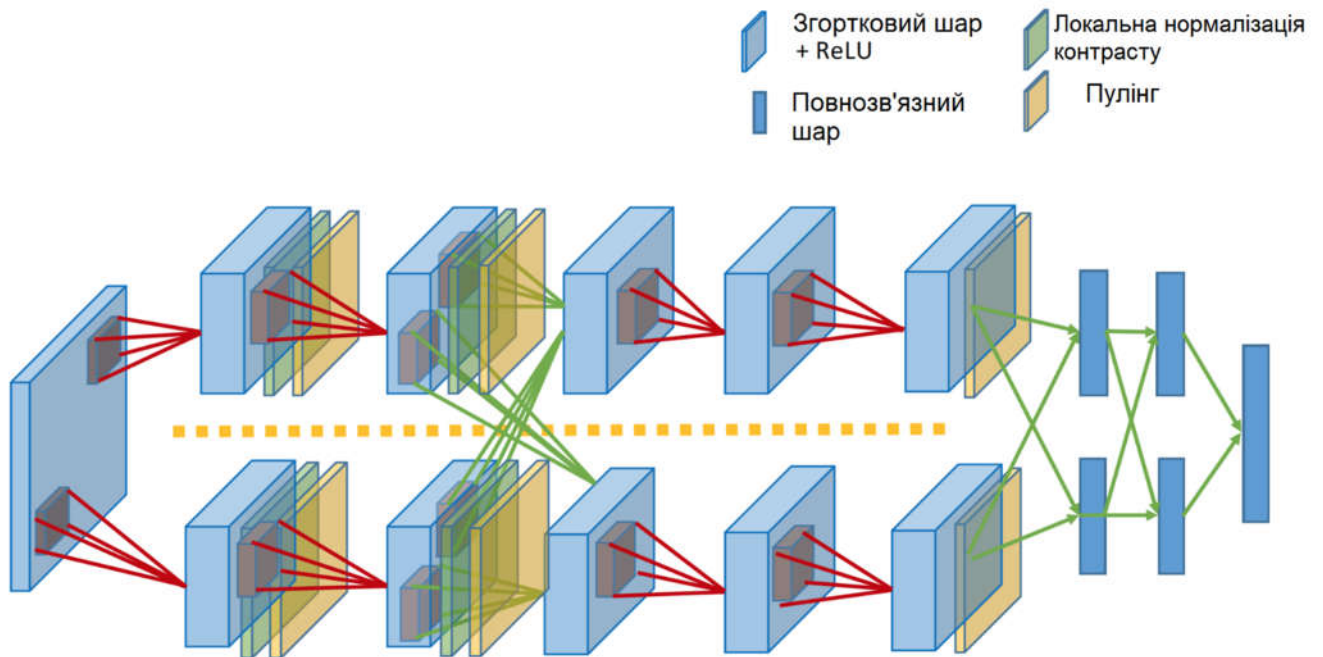


Рисунок 2.2 – Топологія AlexNet для двох GPU

З точки зору топології мережі це майже той же LeNet (рисунок 2.3), просто збільшений в тисячу разів. Додалися ще кілька згорткових шарів, а розмір ядер згортки зменшується від входу мережі до виходу. Це пояснюється тим, що на початку пікселі сильно скорельовані, і рецепторну область можна сміливо брати більшу, ми все одно втрачаємо мало інформації. Далі застосовується пулінг, тим самим збільшуючи щільність некорельованих ділянок. На наступному рівні логічно взяти трохи меншу рецепторну область. В результаті у авторів вийшла така піраміда з згортки $11 \times 11 \rightarrow 5 \times 5 \rightarrow 3 \times 3 \dots$

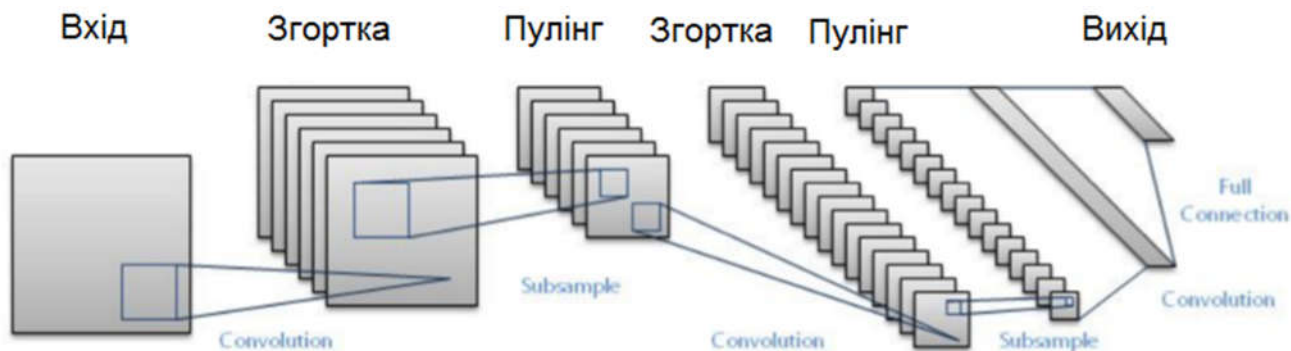


Рисунок 2.3 – Топологія ЗНМ LeNet

В 2013 було створено нову модель згорткових нейронних мереж Меттью Зейлером та Робом Фергюсом з NYU. Названа ZF Net (рисунок 2.4), ця модель досягла 11,2% порогу похибки. Ця архітектура була більш точно налаштована ніж попередня структура AlexNet, але все-таки розроблена з деякими ключовими ідеями щодо підвищення продуктивності. Ще однією причиною того, чому це була така чудова робота, тому що автори провели достатньо часу, дослідивши те, як правильно візуалізувати фільтри та ваги [45].

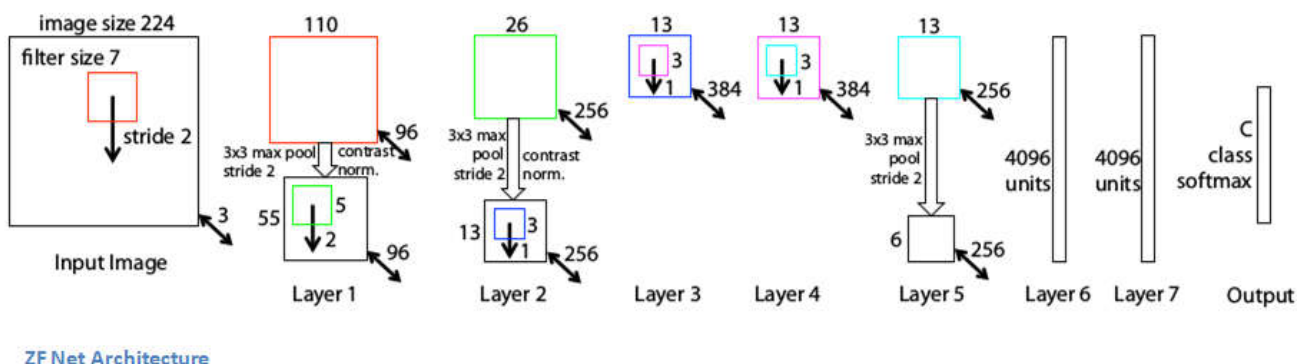


Рисунок 2.4 – Топологія ЗНМ ZF Net

Дуже схожа архітектура до AlexNet, крім декількох незначних модифікацій. AlexNet була тренована на 15 мільйонах зображень, а ZF Net - лише на 1,3 мільйонах зображень.

Замість використання фільтрів розміром 11x11 у першому шарі (саме так, як було реалізовано у AlexNet), ZF Net використовувала фільтри розміром 7x7 [45] і зменшену кількість кроків. Обґрунтування цієї модифікації полягає в тому, що

менший розмір фільтра в першому згортковому шарі допомагає зберігати велику кількість оригінальної інформації про пікселі у вхідному томі [46].

В 2014 році конкурс ILSVRC 2014 виграла згорткова нейронна мережа від компанії Google – GoogLeNet (рисунок 2.5) з коефіцієнтом похибки 6.7% [45]. GoogLeNet була однією з перших архітектур CNN, які дійсно відхилялися від загального підходу простого об'єднання згорткових і субдискретизуючих шарів в послідовній структурі. Автори статті [47] також підкреслили, що ця нова модель приділяє значну увагу використанню пам'яті та енергії.

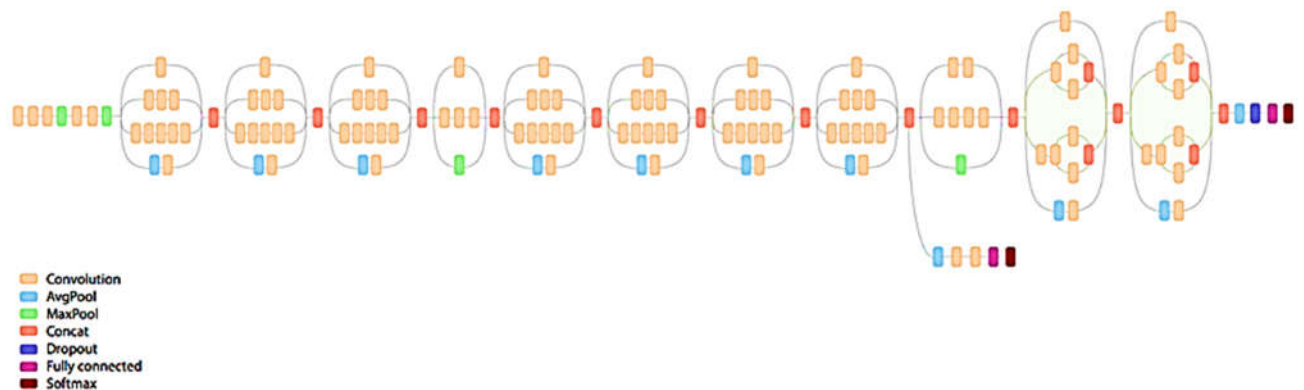


Рисунок 2.5 – Топологія ЗНМ GoogLeNet

Також були застосовані інші методи для уникнення перенавчання, і деякі з них сьогодні (2017) є стандартними для глибоких мереж: DropOut (RIP), Data Augmentation і ReLu [35].

Rethinking the Inception Architecture for Computer Vision – топологія, яка виникла 11 грудня 2015 року [36]. В своїй статті [36] автори дослідили напрактиці різні архітектури і розробили чотири принципи побудови глибоких згорткових нейронних мереж для комп'ютерного зору:

- Уникайте representational bottlenecks: не варто різко знижувати розмірність представлення даних, це потрібно робити плавно від початку мережі і до класифікатора на виході.
- Високорозмірні представлення слід обробляти локально, збільшуючи розмірність: недостатньо плавно знижувати розмірність, варто використовувати принципи для аналізу і групування корельованих ділянок.

- Просторові зв'язки можна і потрібно факторизувати на ще більш дрібні: це дозволить заощадити ресурси і направити їх на збільшення розміру мережі.
- Необхідно дотримуватися балансу між глибиною і шириною мережі: не варто різко збільшувати глибину мережі окремо від ширини, і навпаки; слід рівномірно збільшувати або зменшувати обидві розмірності.

Давно було помічено, що якщо просто вкладати більше шарів, то якість такої моделі зростає до певної межі, а потім починає падати [35]. Цю проблему називають *degradation problem*, а мережі, отримані складанням великої кількості шарів, - *plain*, або плоскі мережі. Дослідники з китайського підрозділу Microsoft змогли знайти таку топологію, при якій якість моделі зростає при додаванні нових шарів. Вона отримала назву ResNet [35]. Дана топологія включає 152 шари.

Крім класичних згорткових нейронних мереж існують також згорткові нейронні мережі, які базуються на регіонах (*Region Based CNNs* [48, 49, 50]).

Поява R-CNNs було більш вражаючою, ніж будь-яка з попередніх статей про нові архітектури ЗНМ. Оскільки перший документ R-CNN [48] був цитований більш ніж у 1600 разів, Росс Гіршик та його група в UC Berkeley здійснили одне з найбільш вражаючих досягнень у комп'ютерному зорі [45]. Як видно з їхніх назв, *Fast R-CNN* [49] та *Faster R-CNN* [50] працювали над тим, щоб зробити модель швидшою і більш підходящою для сучасних задач виявлення об'єктів. На рисунку 2.6 зображено принцип роботи R-CNN мереж.

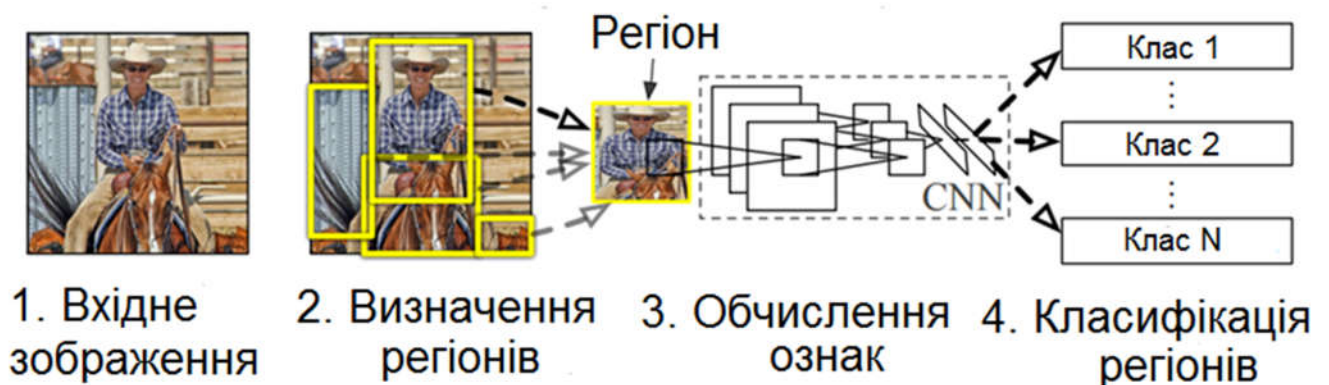


Рисунок 2.6 – Принцип роботи R-CNN мереж

Мета R-CNNs - вирішити проблему виявлення об'єктів. З огляду на певне зображення, ми хочемо мати змогу малювати обмежувальні прямокутники над усіма об'єктами. Процес може бути розділений на два основних компоненти: етап пропозиції регіону та етап класифікації [45].

Для R-CNN, зокрема, використовується вибіркового пошук. Вибірковий пошук виконує функцію генерації 2000 різних регіонів, які з найбільшою ймовірністю містять об'єкт. Після того, як складається комплекс пропозицій регіону, ці пропозиції потім "вирізаються" у розмір зображення, яке може бути подане на вхід CNN (AlexNet в даному випадку), який витягує векторний елемент для кожного регіону. Цей вектор потім використовується як вхідний набір лінійних SVM, які навчаються для кожного класу і виводять класифікацію [48].

2.2. Алгоритми розпізнавання зображень в згорткових нейронних мережах

Згорткові нейронні мережі використовують просторово-місцеву кореляцію шляхом забезпечення локального шаблону зв'язності між нейронами суміжних шарів. Іншими словами, входи прихованих блоків в шарі m взяті з підмножини блоків в шарі $m-1$, що мають просторово суміжні рецептивні поля [36]. Графічно це представлено на рисунку 2.7.

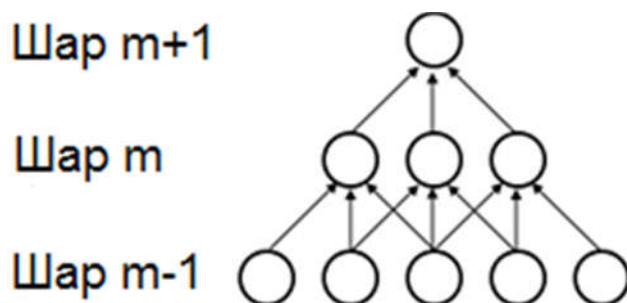


Рисунок 2.7 – Зв'язність між нейронами суміжних шарів

Шар $m-1$ є аналогією до входу сітківки ока. На наведеному вище рисунку (2.1), елементи в шарі m мають рецептивні поля шириною 3 у вхідному шарі, і, таким чином, тільки 3 з'єднані з сусідніми нейронами в наступному шарі. Нейрони в шарі $m+1$ мають аналогічну можливість з'єднання з нейронами наступного шару. Кажуть, що їх сприйнятливим поле щодо наступного шару також 3, але їх сприйнятливим поле по відношенню до входу дорівнює 5 [35]. Кожен блок не реагує на зміни за межами його рецепторного поля по відношенню до сітківки. Таким чином, архітектура гарантує, що витягнуті «фільтри» виробляють найсильнішу відповідь на просторово-локальний характер вхідного сигналу.

Однак, як показано вище, побудова багатьох таких шарів призводить до нелінійних «фільтрів», які стають все більш і більш «глобальними» (тобто реагують на велику область пікселів). Наприклад, нейрон в прихованому шарі $m+1$ може кодувати нелінійну характеристику шириною 5 (в термінах піксельного простору).

В згорткових нейронних мережах кожен фільтр h_i реплікується по всьому зоровому полю. Ці репліковані блоки розподіляють між собою одну й ту ж параметризацію (вектор ваги і зміщення) та утворюють карту ознак (рисунок 2.8).

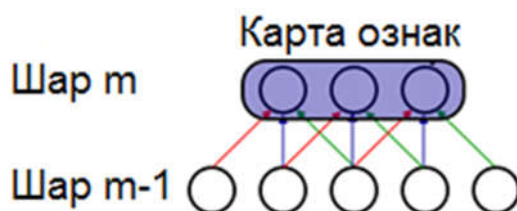


Рисунок 2.8 – Карта ознак шару

Однією з основних переваг згорткових мереж є використання спільної ваги у згорткових шарах, що означає, що для кожного пікселя шару використовується один і той же фільтр (вектор ваги); це як зменшує обсяг необхідної пам'яті, так і поліпшує продуктивність [38].

На рисунку 2.8 показано 3 приховані блоки, що належать до однієї і тієї ж карти ознак. Ваги кожного кольору є спільними. Дана реплікація блоків дозволяє виявляти ключові ознаки незалежно від їхньої позиції в полі зору.

Крім того, спільні ваги підвищують ефективність завдяки зменшенню змінних параметрів мережі. Дані обмеження на моделі дозволяють досягти кращого узагальнення на завданнях комп'ютерного зору.

Карта ознак отримана шляхом багаторазового застосування функції згортки до під-регіонів цілого зображення. Іншими словами, за допомогою згортки вхідного зображення з лінійним фільтром, додаючи зміщення і потім застосовуючи нелінійну функцію.

Якщо ми позначимо k -ту карту ознак заданого шару як h^k , фільтри якої визначаються вагами W^k і зміщенням b_k , то карту ознак h^k одержують наступним чином:

$$h_{ij}^k = \tanh((W^k * x)_{ij} + b_k). \quad (2.1)$$

В порівнянні з іншими алгоритмами класифікації зображень, згорткові нейронні мережі використовують відносно мало попередньої обробки. Це означає, що мережа несе відповідальність за навчання фільтрів, що в традиційних алгоритмах розроблялися вручну. Відсутність залежності у розробці ознак від апріорних знань та людських зусиль є великою перевагою ЗНМ.

Перші два типи шарів (convolutional, subsampling), чергуючись між собою, формують вхідний вектор ознак для багатозарового перцептрона.

Операція згортки для двовимірного сигналу може бути представлена наступною формулою:

$$(f * g)[m, n] = \sum_{k,l} f[m - k, n - l] \cdot g[k, l], \quad (2.2)$$

де f - вихідна матриця зображення,

g - ядро (матриця) згортки.

Для одновимірного сигналу операція згортки задається наступним чином:

$$o[n] = f[n] * g[n] = \sum_{u=-\infty}^{\infty} f[u]g[n - u] = \sum_{u=-\infty}^{\infty} f[n - u]g[u] \quad (2.3)$$

Неформально цю операцію можна описати таким чином - вікном розміру ядра g проходимо з заданим кроком (зазвичай 1) все зображення f , на кожному кроці поелементно множимо вміст вікна на ядро g , результат підсумовується і записується в матрицю результату [40].

При цьому в залежності від методу обробки країв вихідної матриці результат може бути меншим від вихідного зображення (valid), такого ж розміру (same) або більшого розміру (full) (рисунок 2.9).

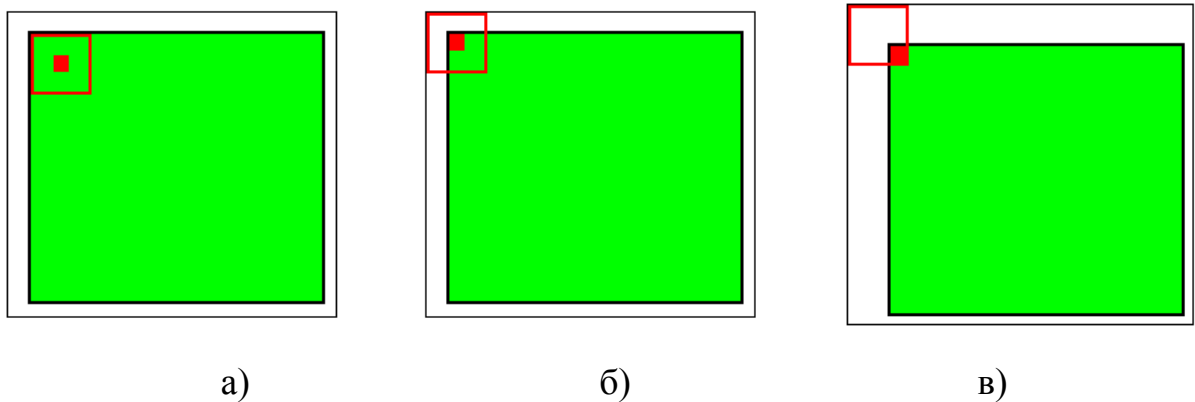


Рисунок 2.9 – обробка країв: а) valid, б) same, в) full

Згортковий шар реалізує ідею так званих локальних рецептивних полів, тобто кожен вихідний нейрон з'єднаний тільки з певною (невеликою) областю вхідної матриці і таким чином моделює деякі особливості людського зору [40]. У спрощеному вигляді цей шар можна описати наступною формулою:

$$x^l = f(x^{l-1} * k^l + b^l), \quad (2.4)$$

де x^l - вихід шару l ,

$f()$ - функція активації,

b - коефіцієнт зсуву,

$*$ - операція згортки входу x з ядром k .

При цьому за рахунок крайових ефектів розмір вихідних матриць зменшується:

$$x_j^l = f(\sum_i x_i^{l-1} * k_j^l + b_j^l), \quad (2.5)$$

де x_j^l - карта ознак j (вихід шару l),

$f()$ - функція активації,

b_j - коефіцієнт зсуву для карти ознак j ,

k_j - ядро згортки номер j ,

x_i^{l-1} - карти ознак попереднього шару.

Варто також відзначити, що хоча згортковий шар скорочує кількість параметрів в порівнянні з повнозв'язним шаром, він використовує більше гіперпараметрів - параметрів, які обирають до початку навчання [37]. На рисунку 2.10 наведено схематичне зображення принципу роботи згорткового шару.

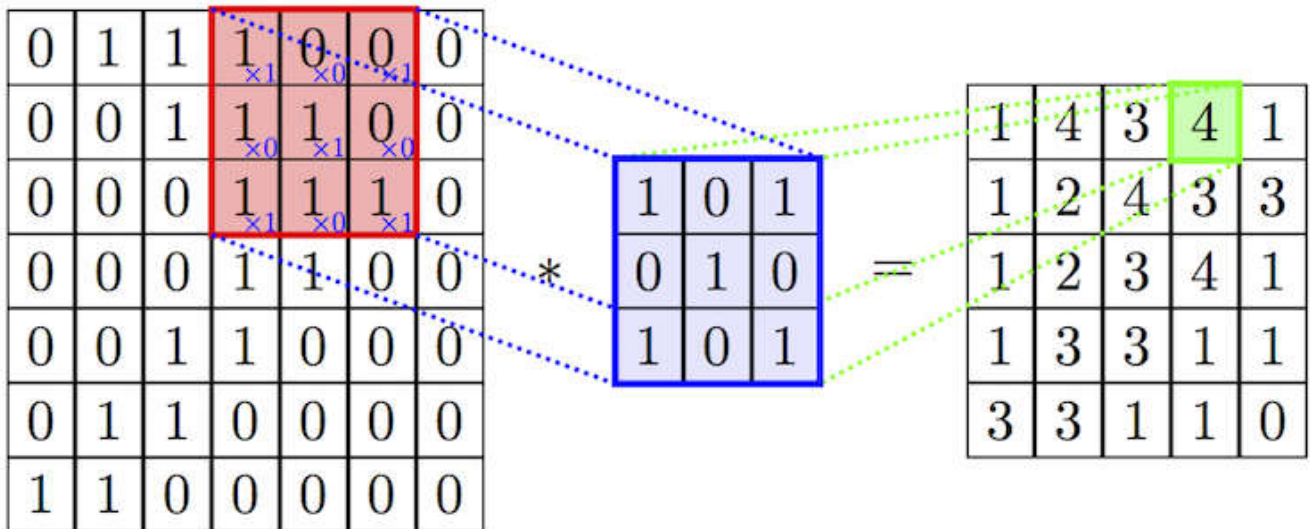


Рисунок 2.10 – Схематичне зображення принципу роботи згорткового шару

Зокрема, вибираються такі гіперпараметри:

- Глибина (depth) - скільки ядер і коефіцієнтів зміщення буде задіяно в одному шарі;
- Висота (height) і ширина (width) кожного ядра;
- Крок (stride) - на скільки зміщується ядро на кожному кроці при обчисленні наступного пікселя результуючого зображення. Зазвичай його приймають рівним 1, і чим більше його значення, тим менший розмір вихідного зображення;
- Відступ (padding): зауважимо, що згортка будь-яким ядром розмірності більше, ніж 1x1 зменшить розмір вихідного зображення. Так як в загальному випадку бажано зберігати розмір початкового зображення, то рисунок доповнюється нулями по краях.

Субдискретизуючі шари в згорткових нейронних мережах виконують зменшення розміру вхідної карти ознак (зазвичай в 2 рази) [40]. Це можна робити різними способами, в даному випадку ми розглянемо метод вибору максимального елемента (max-pooling) - вся карта ознак поділяється на осередки 2x2 елемента, з яких вибираються максимальні за значенням. Тому при субдискретизації зображення значно зменшується у розмірах, але при цьому його зміст не змінюється або змінюється на незначну величину. Формально субдискретизуючий шар може бути описаний таким чином:

$$x^l = f(a^l \cdot \text{subsample}(x^{l-1}) + b^l), \quad (2.6)$$

де x^l - вихід шару l ,

$f()$ - функція активації,

a, b - коефіцієнти,

$\text{subsample}()$ - операція вибірки локальних максимальних значень.

Схематичне зображення операції субдискретизації представлено на рисунку 2.11.

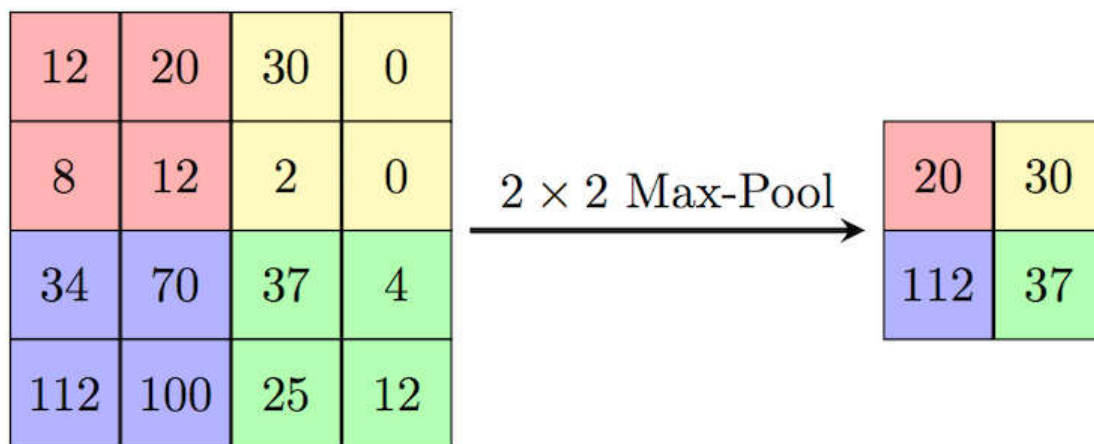


Рисунок 2.11 – Схематичне зображення операції субдискретизації

Останній, але не менш важливий за попередні, тим шарів в згорткових нейронних мережах – повнозв’язний шар (fully connected layer). Вхідні нейрони цього шару під’єднані до всіх активацій в попередньому шарі [41]. Даний шар проводить підрахунок оцінки класу, і на вихід подає вектор, розмірністю N (кількість класів). Таким чином, повнозв’язний шар представляє модель багат шарового перцептрона (MLP), який можна описати таким співвідношенням:

$$x_j^l = f(\sum_i x_i^{l-1} \cdot w_{ij}^{l-1} + b_j^{l-1}), \quad (2.7)$$

де x_j^l - вихід шару l ,

$f()$ - функція активації,

b - коефіцієнт зсуву,

w - матриця вагових коефіцієнтів.

Після цього, проаналізувавши вихідний вектор повнозв’язного шару, результатом роботи згорткової нейронної мережі буде той клас, оцінка якого виявилась найбільшою у даному векторі.

Типова CNN-архітектура (рисунок 2.12) для класифікації зображень може бути розділена на дві різні частини - ланцюжок повторюваних Conv → Pool шарів (іноді з кількома згортковими шарами одночасно), за якими слідує кілька

повнозв'язних шарів (беручи кожен піксель обчислюваних зображень як незалежний вхід), закінченням якого є шаблон soft max, який оптимізовано для крос-ентропії [44].

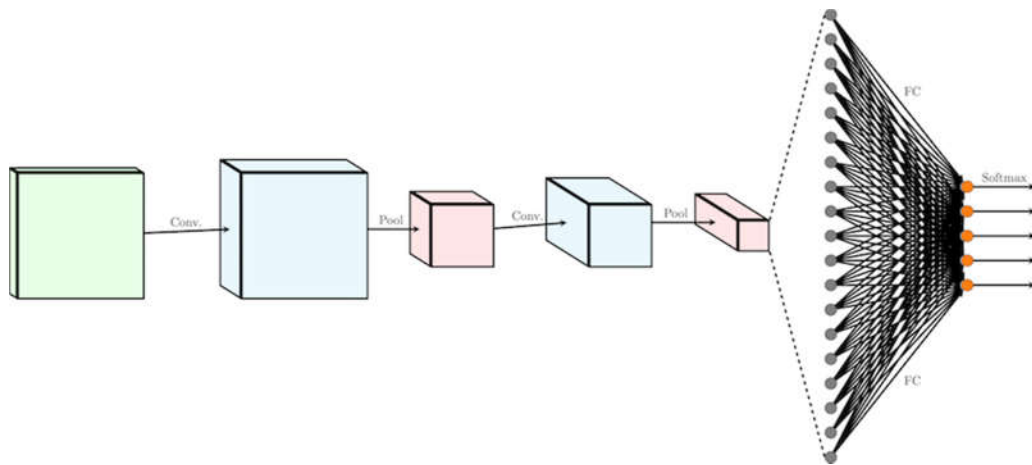


Рисунок 2.12 – Типова архітектура ЗНМ

Зазвичай після кожного згорткового або повнозв'язного шару функція активації (наприклад, ReLU) буде застосована до всіх виходів [44].

2.3. Алгоритми навчання згорткових нейронних мереж

Навчання нейронної мережі - це процес, в якому параметри нейронної мережі настраюються за допомогою моделювання середовища, в яку ця мережа вбудована. Тип навчання визначається способом підстроювання параметрів. Розрізняють алгоритми навчання з вчителем і без вчителя.

Процес навчання з учителем є пред'явлення мережі вибірки навчальних прикладів. Кожен зразок подається на входи мережі, потім проходить обробку всередині структури НС, обчислюється вихідний сигнал мережі, який порівнюється з відповідним значенням цільового вектора, що представляє собою необхідний вихід мережі [51].

Для того, щоб нейронна мережі була здатна виконати поставлене завдання, її необхідно навчити (рисунок 2.13). Розрізняють алгоритми навчання з вчителем і без вчителя.

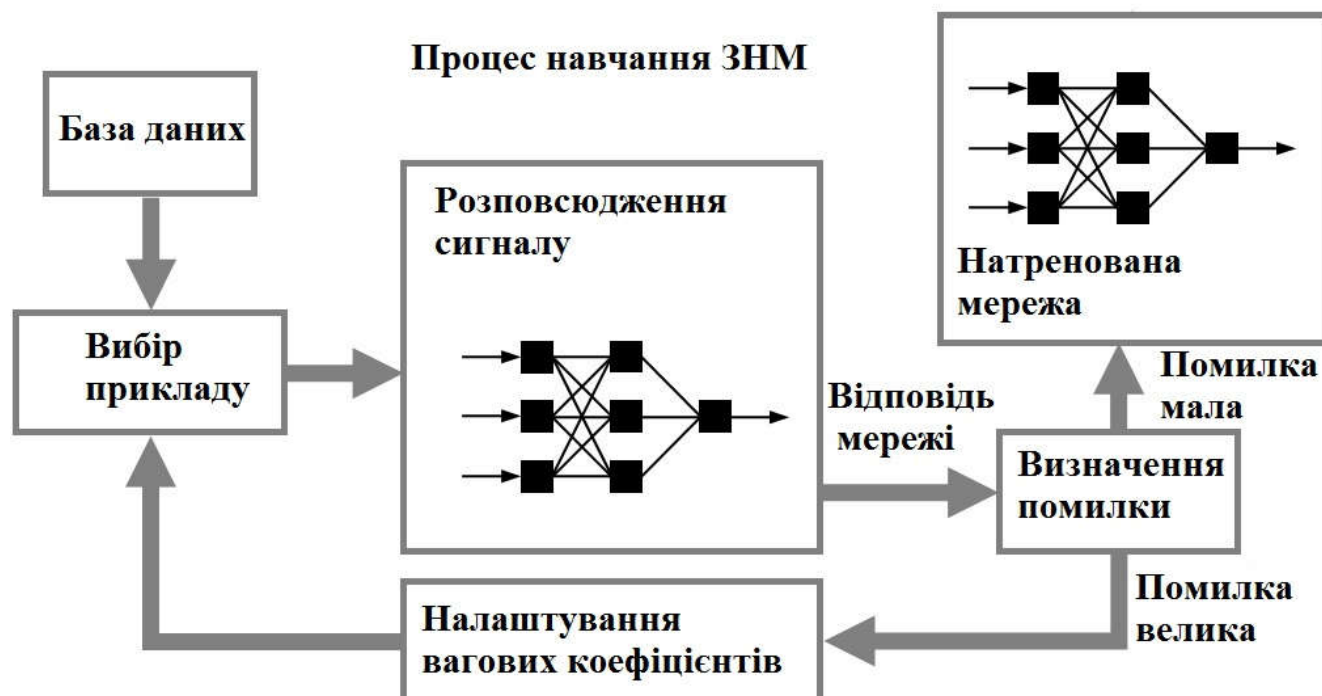


Рисунок 2.13 – Процес навчання згорткової нейронної мережі

Процес навчання з учителем є пред'явлення мережі вибірки навчальних прикладів. Кожен зразок подається на входи мережі, потім проходить обробку всередині структури НС, обчислюється вихідний сигнал мережі, який порівнюється з відповідним значенням цільового вектора, що представляє собою необхідний вихід мережі [51]. Потім за певним правилом обчислюється помилка, і відбувається зміна вагових коефіцієнтів зв'язків всередині мережі в залежності від обраного алгоритму. Вектори навчальної множини пред'являються послідовно, обчислюються помилки і ваги підлаштовуються для кожного вектора до тих пір, поки помилка по всьому навчальному масиву не досягне прийнятного рівня.

При навчанні без учителя навчальну множину складається лише з вхідних векторів. Навчальний алгоритм підлаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення досить близьких вхідних

векторів давало однакові виходи. Процес навчання, отже, виділяє статистичні властивості навчальної множини і групує подібні вектори в класи. Пред'явлення на вхід вектора з даного класу дасть певний вихідний вектор, але до навчання неможливо передбачити, який вихід буде проводитися даним класом вхідних векторів. Отже, виходи подібної мережі повинні трансформуватися в деяку зрозумілу форму, зумовлену процесом навчання. Це не є серйозною проблемою. Зазвичай не складно ідентифікувати зв'язок між входом і виходом, встановлену мережею.

Для навчання нейронних мереж без вчителя застосовуються сигнальні метод навчання Хебба і Ойа.

Математично процес навчання можна описати таким чином. В процесі функціонування нейронна мережа формує вихідний сигнал Y , реалізуючи деяку функцію $Y = G(X)$. Якщо архітектура мережі задана, то вид функції G визначається значеннями синаптичних ваг і зміщеною мережі.

Нехай рішенням деякої задачі є функція $Y = F(X)$, задана параметрами вхідних-вихідних даних $(X^1, Y^1), (X^2, Y^2), \dots, (X^N, Y^N)$, для яких $Y^k = F(X^k)$ ($k = 1, 2, \dots, N$).

Навчання складається в пошуку (синтезі) функції G , близькою до F в сенсі деякої функції помилки E .

Якщо вибрано множину навчальних прикладів - пар (X^k, Y^k) ($de k = 1, 2, \dots, N$) і спосіб обчислення функції помилки E , то навчання нейронної мережі перетворюється в задачу багатовимірної оптимізації, що має дуже велику розмірність, при цьому, оскільки функція E може мати довільний вигляд навчання в загальному випадку - багатоекстремальна неопукла задача оптимізації [51].

Для вирішення цієї задачі можуть використовуватися такі (ітераційні) алгоритми:

1. Алгоритми локальної оптимізації з обчисленням часткових похідних першого порядку:

- градієнтний алгоритм (метод найшвидшого спуску),

- методи з одновимірною і двовимірною оптимізацією цільової функції в напрямку антиградієнту,
- метод спряжених градієнтів,

2. Методи, що враховують напрямок антиградієнту на кілька кроків алгоритму;

- алгоритми локальної оптимізації з обчисленням приватних похідних першого і другого порядку:

- метод Ньютона,
- методи оптимізації з розрідженими матрицями Гессе,
- квазіньютонівські методи,
- метод Гаусса-Ньютона,
- метод Левенберга-Марквардта;

3. Стохастичні алгоритми оптимізації:

- пошук у випадковому напрямку,
- метод Монте-Карло (чисельний метод статистичних випробувань);

4. Алгоритми глобальної оптимізації (задачі глобальної оптимізації вирішуються за допомогою перебору значень змінних, від яких залежить цільова функція).

Найбільш простим і популярним способом навчання є метод навчання з учителем (на маркованих даних) - метод зворотного поширення помилки і його модифікації.

Для того, щоб можна було почати навчання згорткової нейронної мережі потрібно визначитися з тим, як вимірювати якість розпізнавання. Найпоширенішим методом вимірювання в теорії нейронних мереж є функція середньоквадратичної помилки (MSE) [42]:

$$E^p = \frac{1}{2} (D^p - O(I^p, W))^2, \quad (2.8)$$

де E^p - помилка розпізнавання для p -ої навчальної пари,

D^p - бажаний вихід мережі,

$O(I^p, W)$ - вихід мережі, що залежить від p -го входу і вагових коефіцієнтів W , куди входять ядра згортки, зміщення, вагові коефіцієнти.

Завдання навчання так налаштувати ваги W , щоб вони для будь-якої навчальної пари (I^p, D^p) давали мінімальну помилку E^p . Щоб порахувати помилку для всієї навчальної вибірки - береться середнє арифметичне по помилках для всіх навчальних пар [42]. Для мінімізації функції помилки E^p найефективнішими є градієнтні методи.

Помилка на виході шару субдискретизації розраховується шляхом виконання "зворотної згортки" карт ознак наступного за ним згорткового шару, тобто над кожною картою ознак виконується згортка з відповідним "перевернутим" ядром, при цьому за рахунок крайових ефектів розмір вихідних матриць збільшується [40]. Далі над отриманими картами обчислюються кілька часткових сум по числу ядер згортки, відповідно до матриці суміжності субдискретизуючого і згорткового шарів.

Навчання алгоритмом зворотного поширення помилки передбачає два проходи по всім шарам мережі: прямого і зворотного. При прямому проході вхідний вектор подається на вхідний шар нейронної мережі, після чого поширюється по мережі. Під час прямого проходу всі синаптичні ваги мережі фіксовані [43]. Під час зворотного проходу всі синаптичні ваги налаштовуються відповідно до правила корекції помилок, а саме: фактичний вихід мережі віднімається з бажаного, в результаті чого формується сигнал помилки. Цей сигнал згодом поширюється по мережі в напрямку, протилежному напрямку синаптичних зв'язків.

Алгоритм зворотного поширення помилки наступний:

- 1) Ініціалізувати синаптичні ваги маленькими випадковими значеннями.
- 2) Вибрати чергову навчальну пару з навчальної множини; подати вхідний вектор на вхід мережі.
- 3) Обчислити вихід мережі.
- 4) Обчислити різницю між виходом мережі і необхідним виходом (цільовим вектором навчальної пари).

- 5) Підкоригувати ваги мережі для мінімізації помилки.
- 6) Повторювати кроки з 2 по 5 для кожного вектора навчальної множини доти, поки помилка на всій множині не досягне прийнятного рівня.

Операції, що виконуються кроками 2 і 3, подібні до тих, які виконуються при функціонуванні вже навченої мережі, тобто подається вхідний вектор і обчислюється вихід для нього. Обчислення виконуються пошарово.

Важливим параметром в процесі навчання згорткової нейронної мережі є функція активації. Функція активації - це спосіб нормалізації вхідних даних. Тобто, якщо на вході нейрону буде велике число, пропустивши його через функцію активації, ми отримаємо значення в потрібному діапазоні. Функцій активації існує досить багато, але найпоширенішими є: Лінійна (рисунок 2.14), сигмоїд (рисунок 2.15) і Гіперболічний тангенс (рисунок 2.16) [52]. Головні їх відмінності - це діапазон значень.

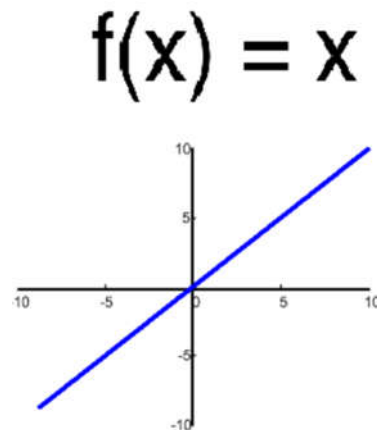


Рисунок 2.14 – Лінійна функція активації

У випадку лінійної функції активації сигнал на виході нейрона лінійно пов'язаний зі зваженою сумою сигналів на його вході. У штучних нейронних мережах з шаруватою структурою нейрони з передавальними функціями такого типу, як правило, складають вхідний шар.

$$f(x) = \frac{1}{1+e^{-x}}$$

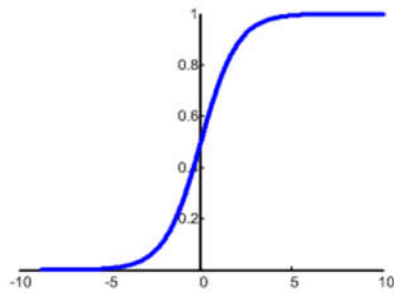


Рисунок 2.15 – Сигмоїдальна функція активації

Сигмоїдальна функція активації є найпоширенішою, її діапазон значень $[0,1]$. Саме на ній побудовані більшість сучасних ЗНМ, також її іноді називають логістичною функцією. Відповідно, якщо в вибірці даних випадку присутні негативні значення, то знадобиться функція, яка захоплює і негативні значення.

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

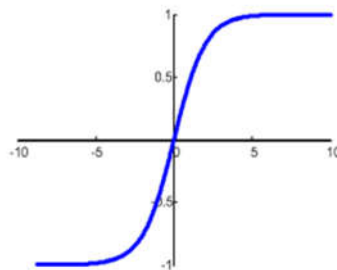


Рисунок 2.16 – Гіперболічний тангенс

Має сенс використовувати гіперболічний тангенс, тільки тоді, коли ваші значення можуть бути і негативними, і позитивними, так як діапазон функції $[-1,1]$. Використовувати цю функцію тільки з позитивними значеннями недоцільно так як це значно погіршить результати нейромережі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ РОЗПІЗНАВАННЯ ГІСТОЛОГІЧНИХ ЗОБРАЖЕНЬ

3.1. Функційна структура програмного модуля

Розроблене програмне забезпечення складається з наступних частин: головний модуль, модуль тестування згорткової нейронної мережі, модуль типу моделі, модуль навчання нейронної мережі та модуль візуалізації роботи ЗНМ. Крім цього, в структуру програмного модуля розпізнавання гістологічних зображень входить набір fxml-файлів графічного інтерфейсу. Структурна схема розробленого програмного модуля наведена на рисунку 3.1.

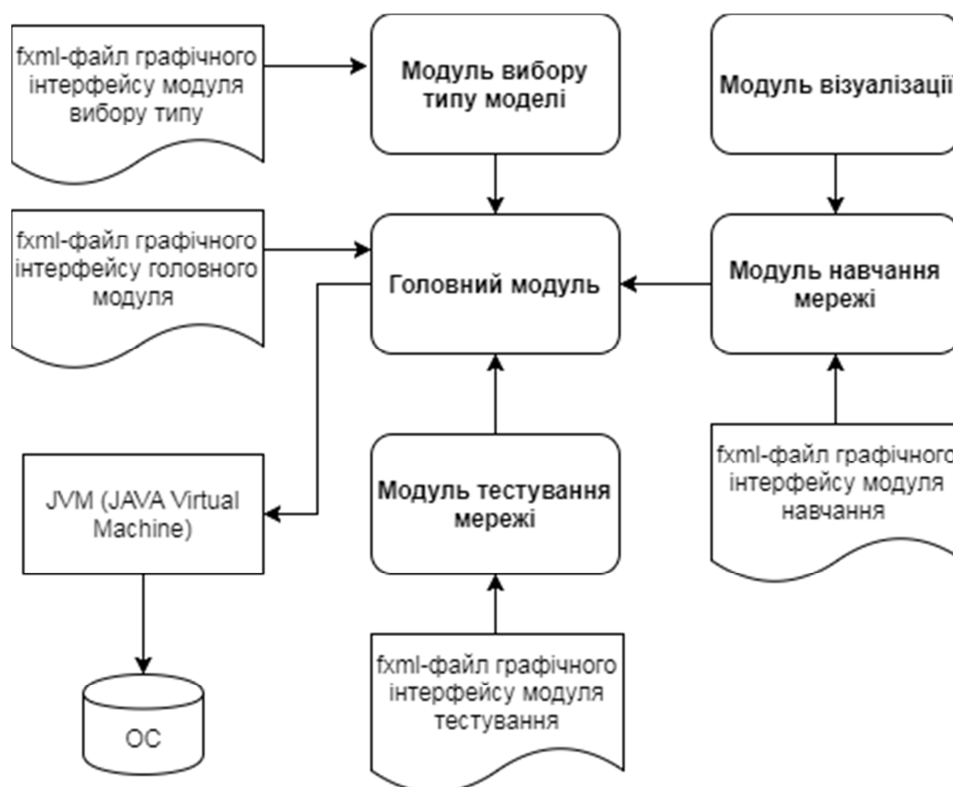


Рисунок 3.1 - Структурна схема розробленого програмного модуля

Головний модуль виконує функції ядра для всіх інших модулів, зв'язуючи їх в єдине ціле та являє собою єдину точку входу в програму.

Модуль тестування мережі призначений для вибору файлу попередньо натренованої моделі, вибору зображення, яке потрібно класифікувати, та

безпосередньої класифікації вибраного зображення. З його допомогою користувач має можливість провести тестування своєї мережі та зробити висновки про якість класифікації нею зображень. Блок-схема алгоритму роботи модуля тестування ЗНМ наведена на рисунку 3.2.



Рисунок 3.2 – Блок-схема алгоритму роботи модуля тестування згорткової нейронної мережі

Модуль вибору типу моделі - це перший крок до створення власної моделі. Він надає можливість користувачу вибрати вже існуючу архітектуру згорткової нейронної мережі (AlexNet, Lenet), яку він збирається навчати, або ж вибрати власну архітектуру (Custom). Після вибору архітектури моделі - керування процесом її навчання переходить до модуля навчання мережі.

Модуль навчання згорткової мережі надає широкі можливості до конфігурування вхідних параметрів мережі, а також самого процесу навчання. Сюди входять завантаження навчальної та тестової вибірок, підбір параметрів мережі (функція активації на кожному шарі, поріг навчання) та конфігурація

тривалості навчання, кількості ітерацій і т.д. Блок-схема алгоритму роботи модуля навчання згорткової нейронної мережі наведена на рисунку 3.3.

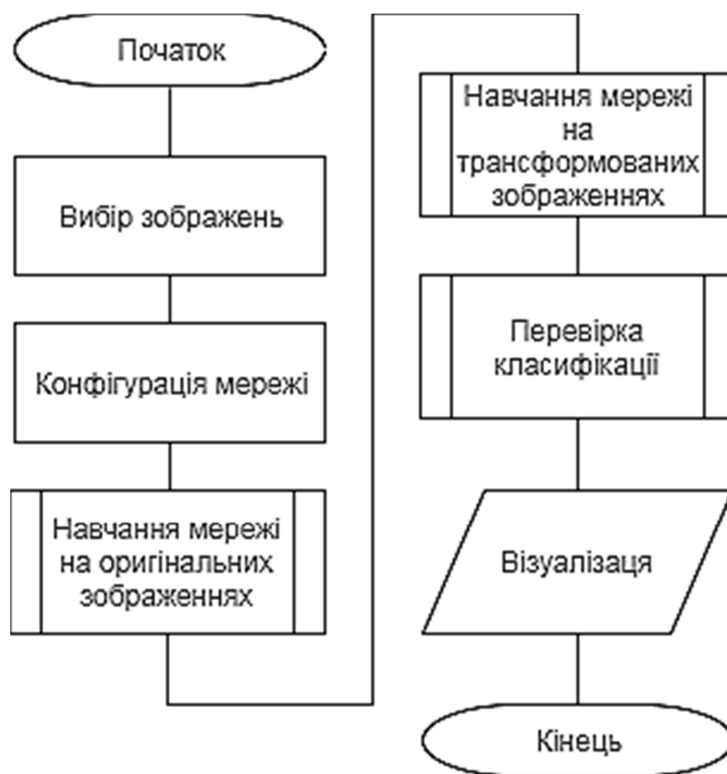


Рисунок 3.3 – Блок-схема алгоритму роботи модуля навчання згорткової нейронної мережі

Крім цього, модуль навчання надає можливість зберегти натреновану модель в файл і завантажити його під час тестування для класифікації тестового зображення.

Модуль візуалізації дозволяє переглянути всю інформацію про згорткову нейронну мережу, про кожен її шар та її властивості. Крім цього, він надає можливість моніторити процес навчання мережі в режимі реального часу. Даний модуль дозволяє відслідковувати множину параметрів під час тренування нейронної мережі. Це дозволяє проаналізувати їх і використати для отримання інформації про різні гіперпараметри мережі та про те, як їх слід змінювати для більш ефективного навчання.

Графічний інтерфейс реалізований за допомогою технології JavaFX, яка дозволяє створювати програми з насиченим графічним користувацьким

інтерфейсом для безпосереднього запуску з-під операційних систем, роботи в браузері чи на мобільному пристрої.

В розробленій програмі присутні чотири вікна, які описані за допомогою fxml-файлів, структура яких задає позиції та порядок розташування на екрані, а також взаємодію з контроллерами.

Запуск розробленого програмного модуля можливий на будь-якій операційній системі, яка підтримує JVM (Java virtual machine). JVM виконує байт-код Java, який попередньо створюється компілятором. Для коректної роботи програмного модуля необхідне таке додаткове системне програмне забезпечення: Java версії не нижче 1.8.0_45.

Мінімальні вимоги до апаратного забезпечення наступні: RAM 2GB, CPU 2 core 2.3GHz, Місця на диску: 200Mb, Розрядність ОС: 64.

3.2. UML-діаграми та опис реалізації

UML - уніфікована мова моделювання програмних систем, яка використовується у парадигмі об'єктно-орієнтованого програмування. UML включає в себе множину типів діаграм: структурні діаграми (класів, компонент, об'єктів, розгортання), діаграми поведінки (діяльності, скінченних автоматів, прецедентів), діаграми взаємодії та ін.

3.2.1 Діаграма прецедентів

Для початку побудуємо діаграму прецедентів (Use Case diagram) програмного продукту (рисунок 3.4).

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед

прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.



Рисунок 3.4 – Діаграма прецедентів

Як видно з рисунку 3.4, користувач може взаємодіяти з програмною системою наступним чином: вибрати тип моделі, завантажити навчальну і тестову вибірку зображень, здійснити конфігурацію мережі, провести її навчання, зберегти файл натренованої моделі на диск, провести візуалізацію. Крім цього користувач може здійснити тестування згорткової нейронної мережі шляхом класифікації тестового зображення та вийти з програми.

3.2.2 Діаграма послідовностей

Діаграма послідовностей відноситься до діаграм взаємодії UML, що описує поведінкові аспекти системи, але розглядає взаємодію об'єктів в часі. На рисунку 3.6 представлено діаграму послідовностей для сценарію фільтрації зображення.

На діаграмі послідовностей показано у вигляді вертикальних ліній різні процеси або об'єкти, що існують водночас. Надіслані повідомлення зображуються у вигляді горизонтальних ліній, в порядку відправлення. Визначені стандартом

UML 2.0 діаграми послідовностей мають ті ж можливості що і визначені стандартом UML 1.x, та підтримують додаткові можливості зміни стандартного порядку повідомлень.

На рисунку 3.5 наведено діаграму послідовностей для сценарію процесу навчання згорткової нейронної мережі.



Рисунок 3.5 – Діаграма послідовностей для сценарію навчання згорткової нейронної мережі

Як видно з даної діаграми, в системі існує п'ять об'єктів: Controller, Training Session, Training Helper, Model, UILogger. Кожен з них виконує певні функції над даними з попереднього об'єкту і передає опрацьовані дані наступному об'єкту.

3.2.3 Діаграма пакетів

Діаграми пакетів відображають залежності між пакетами, з яких складається програмне забезпечення. На рисунку 3.6 наведено діаграму пакетів для розробленого програмного модуля. Пакет Controller імпортує класи пакету Tools. Інші відношення між пакетами означають їх залежність один від одного.

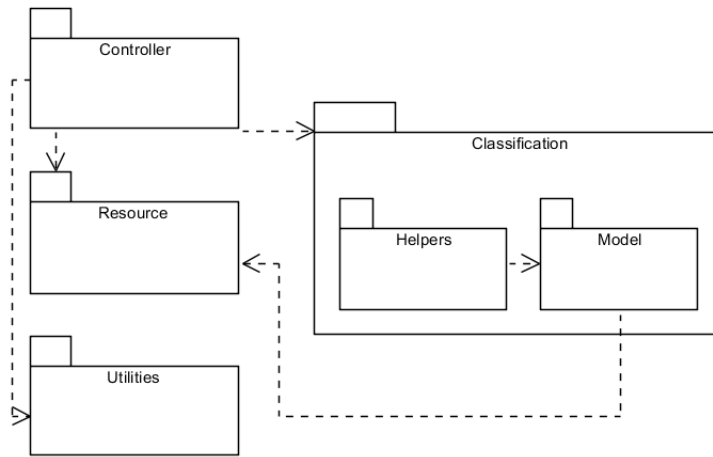


Рисунок 3.6 – Діаграма пакетів

На рисунку 3.7 наведено структуру проекту з точки зору структуризації та поділу на пакети і класи.

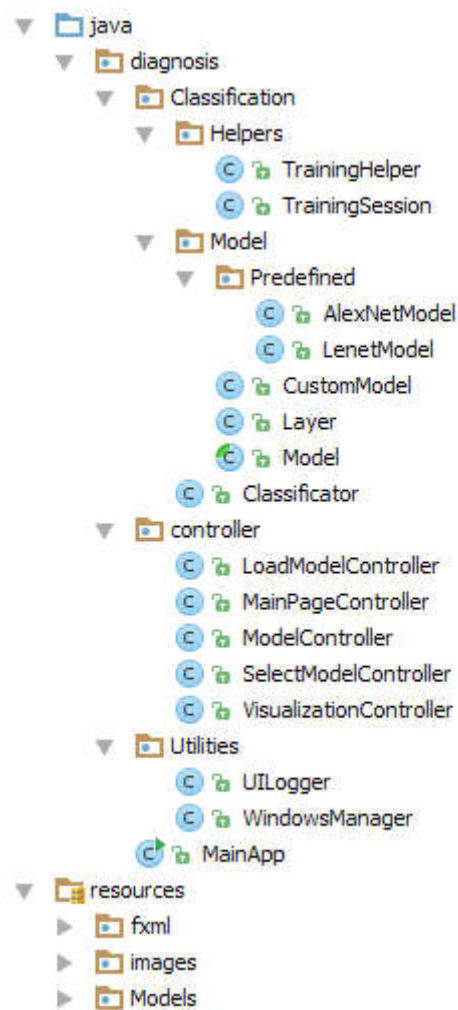


Рисунок 3.7 – Структура проекту в середовищі IntelliJ IDEA

3.2.4 Діаграма класів

Діаграма класів - це набір статичних, декларативних елементів моделі. Діаграми класів можуть застосовуватися і при прямому проектуванні, тобто в процесі розробки нової системи, і при зворотному проектуванні - описі існуючих і використовуваних систем. В додатку А наведено діаграму класів розроблюваного програмного продукту.

Як видно з діаграми класів (див. додаток Б), програмний модуль складається з наступних класів: MainApp, LoadModelController, MainPageController, ModelController, SelectModelController, VisualizationController, TrainingHelper, TrainingSession, Layer, Model, AlexNetModel, LenetModel, CustomModel, UILogger, WindowsManager.

Як видно з наведеної діаграми класів, клас Model наслідує базовий клас Layer. В класі Layer реалізовані методи для побудови шарів згорткової нейронної мережі заданих параметрів та типів. Клас Model успадковує дані методи та реалізує методи, які вже безпосередньо з ними працюють. Даний клас є абстрактним. Це означає, що об'єкт цього класу створити не можна, а можна тільки наслідувати.

Класи AlexNetModel, LenetModel і CustomModel розширяють клас Model, реалізуючи цим самим відповідні моделі і властивою їм архітектурою і всіма функціями для роботи з ними.

Клас TrainingSession призначений для зберігання параметрів процесу навчання, навчальної та тестової вибірки та ін. Він передається в якості конструктора об'єкту класу TrainingHelper. Даний клас виходячи з переданого йому типу моделі, створює об'єкт моделі відповідного класу і запускає процес навчання відповідно до заданих параметрів навчання і конфігурації згорткової нейронної мережі.

Крім цього, існують допоміжні класи (WindowsManager та UILogger), які спрощують взаємодію з вищеописаними класами. Перший відповідає за роботу з вікнами (створення, відкриття та ін.) на кожному з етапів роботи програмного

модуля. Другий – за журналювання процесу роботи, навчання мережі та виваедення результатів на екран.

3.3. Програмний модуль розпізнавання гістологічних зображень

Програмний модуль класифікації гістологічних зображень реалізований з використанням мови програмування Java, технології JavaFX, середовища розробки IntelliJ IDEA та шаблону проектування «модель-вид-контроллер» (MVC).

MVC передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону - гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

Оскільки програмний модуль розроблено з використанням мови програмування Java, то це дає можливість запускати його на будь-якій операційній системі, яка підтримує технологію JVM (Java Virtual Machine). А це практично всі сучасні операційні системи, включаючи мобільні платформи.

Після запуску програмного модуля відкривається головне вікно, яке містить дві кнопки: «Create New Model» та «Load Existing Model». Перша дозволяє перейти на наступне вікно, де буде запропоновано вибрати архітектуру моделі (AlexNet, Lenet або власну - Custom). Друга кнопка надає можливість перейти на вікно тестування мережі, де можна вибрати з диску файл існуючої моделі та зображення і здійснити його класифікацію. На рисунку 3.8 наведено головне

вікно розробленого програмного модуля та вікно вибору архітектури моделі нейронної мережі.

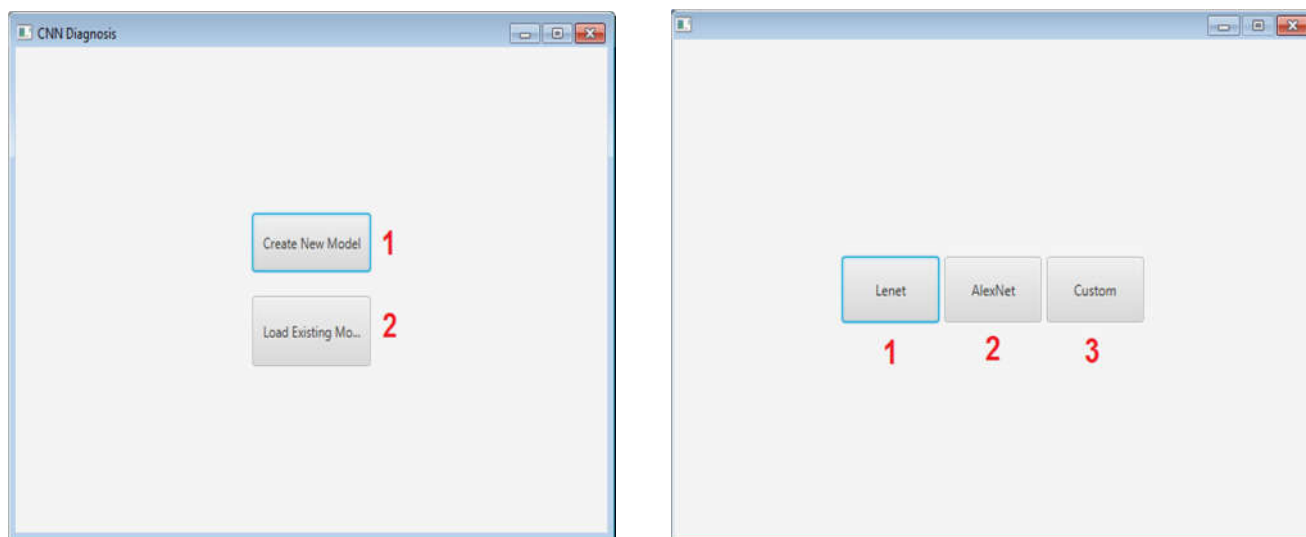


Рисунок 3.8 – Головне вікно та вікно вибору архітектури моделі нейронної мережі

Після того, як користувач вибрав архітектуру моделі, відкриється вікно конфігурації процесу навчання та вибору її вхідних параметрів (рисунок 3.9).

На рисунку 3.9 цифрами позначено:

1. Головне меню;
2. шлях до папки з навчальною та тестовою вибірками;
3. кнопка, яка відкриває діалог вибору директорії з навчальною та тестовою вибірками;
4. компонент для підбору кількості епох навчання мережі;
5. компонент для підбору кількості ітерацій навчання мережі;
6. група компонентів, призначена для вибору типів трансформацій зображень під час навчання;
7. поле для вказання порогу навчання мережі;
8. компонент вибору функції активації;
9. кнопка для старту процесу навчання;
10. кнопка для відкривання вікна з візуалізацією процесу навчання моделі;
11. область додаткової інформації про процес навчання і роботи мережі.

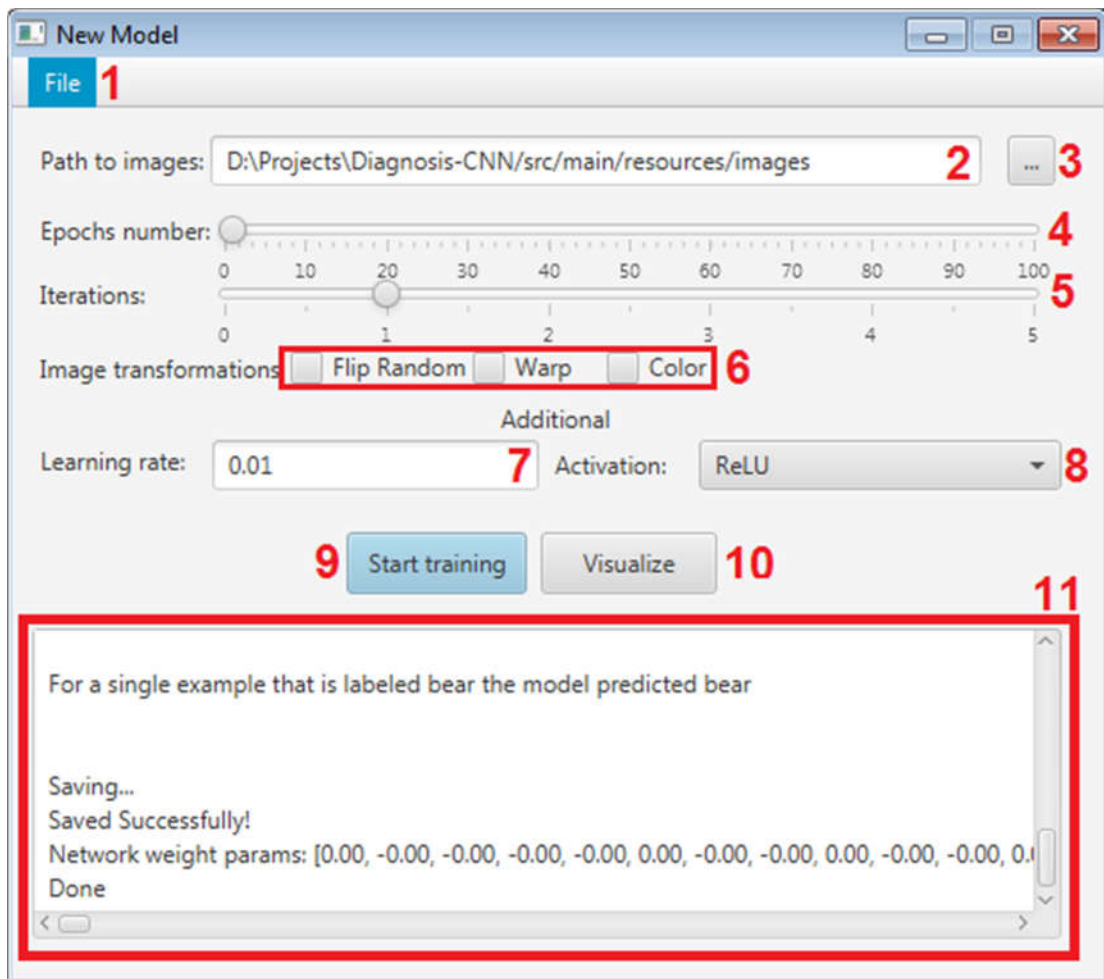


Рисунок 3.9 - Вікно конфігурації процесу навчання та вибору її вхідних параметрів

Елементи 4 та 5 являють собою компоненти, які дозволяють вибрати кількість епох та ітерацій процесу навчання згорткової нейронної мережі шляхом пересування відповідних повзунків (рисунок 3.10). Дозволений діапазон кількості епох: 0-100. Дозволений діапазон кількості ітерацій: 0-5.



Рисунок 3.10 – Компоненти для вибору кількості епох та ітерацій навчання мережі

На рисунку 3.11 представлено компонент для вибору виду трансформацій навчальних зображень під час тренування моделі.

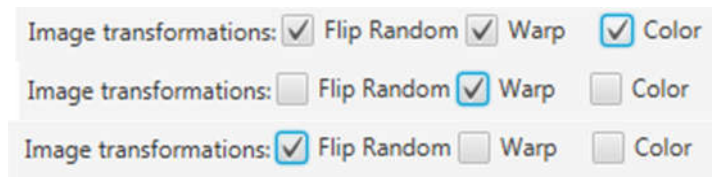


Рисунок 3.11 – Компонент для вибору виду трансформацій навчальних зображень під час тренування моделі та деякі з можливих комбінацій

Як видно з рисунку 3.11 – існує три типи трансформацій зображення під час навчання мережі. А саме: Flip Random – випадкове перевертання, Warp – деформація, Color – випадкова зміна кольорів. Із застосуванням даних трансформацій ми отримуємо більш різноманітну тестову вибірку і як результат – вищу якість навчання.

Крім трансформацій, існують можливість вибрати функцію активації після згорткового шару. Доступні варіанти: ReLU, Sigmoid, ELU, Leaky ReLU, Softmax, Softplus, SoftSign, Tanh.

Функція активації - це спосіб нормалізації вхідних даних. Функцій активації досить багато, але найбільш популярними є: Лінійна, Сигмоїд (Логістична) і Гіперболічний тангенс. Головні їх відмінності - це діапазон значень. Компонент вибору функції активації зображено на рисунку 3.12.

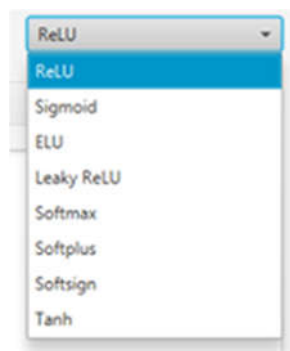


Рисунок 3.12 – Компонент для вибору функції активації та можливі варіанти вибору

Розроблений програмний модуль надає широкі можливості для візуалізації процесу навчання мережі та всієї інформації про кожен з її шарів. Для перегляду даної інформації необхідно натиснути кнопку 10 (рисунок 3.9).

DL4J Надає користувацький інтерфейс для візуалізації в браузері (в режимі реального часу) поточного статусу мережі та прогресу навчання. Інтерфейс користувача зазвичай використовується для налаштування нейронних мереж, тобто вибору гіперпараметрів (наприклад, швидкість навчання) для отримання високої продуктивності для мережі.

Вікно візуалізації (рисунок 3.13) розділене на 4 підменю:

1. загальна інформація про процес навчання, значення функції втрат, співвідношення параметрів до оновлень (за шаром) для всіх вагових коефіцієнтів мережі та ітерації; стандартні відхилення (від часу): активації, градієнти та оновлення;
2. сторінка моделі містить граф шарів нейромережі, який працює як механізм вибору. Щоб відобразити детальну інформацію про шар мережі, необхідно натиснути на нього;
3. інформація про програмне та апаратне забезпечення (навантаженість під час навчання і т.д.);
4. меню вибору мови.

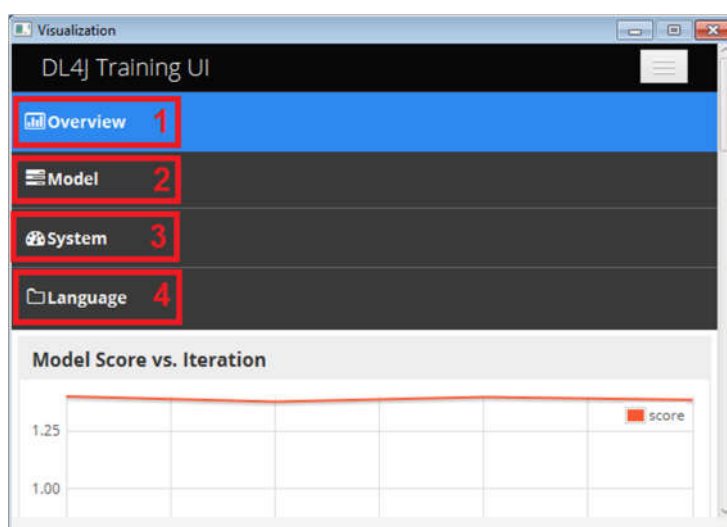


Рисунок 3.13 – Меню вікна візуалізації процесу навчання мережі

Розглянемо кожен з вищенаведених пунктів детальніше. На рисунку 3.14 наведено графік зміни функції втрат в процесі навчання нейромережі.

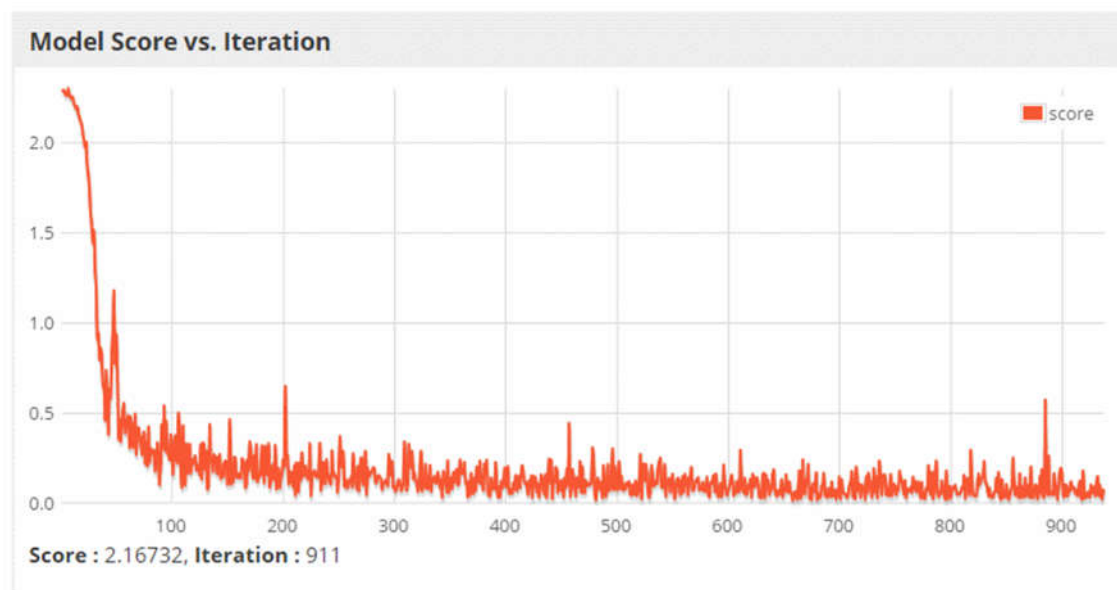


Рисунок 3.14 – Графік зміни функції втрат в процесі навчання

Значення функції втрат повинне постійно зменшуватись в процесі навчання. Якщо воно послідовно зростає, то швидкість навчання, ймовірно, буде занадто високою. Необхідно спробувати зменшити його, поки показники стануть більш стабільними.

Збільшення даного значення також може свідчити про інші проблеми в мережі, такі як неправильна нормалізація даних.

На рисунку 3.15 наведено приклад представлення загальної інформації про процес навчання.

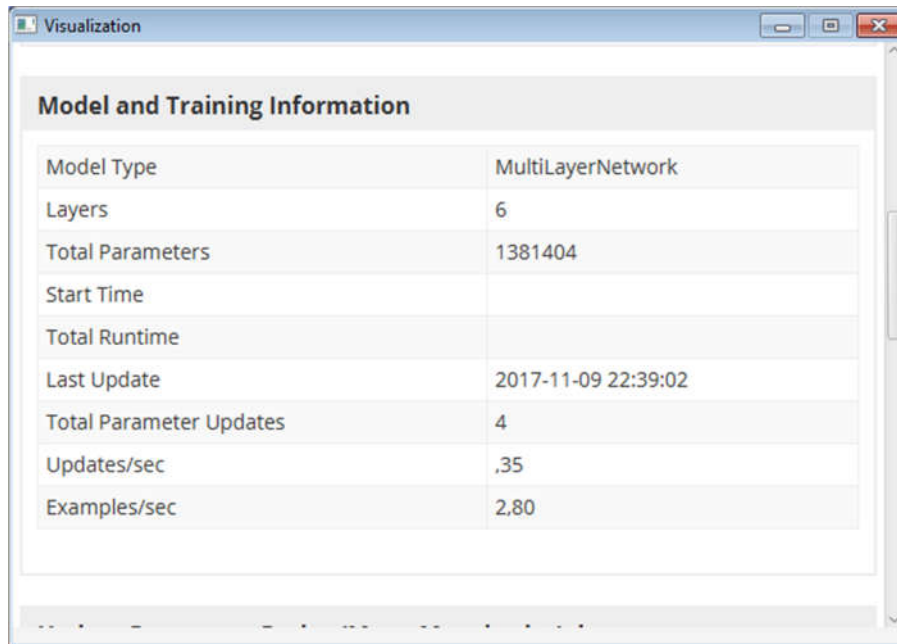
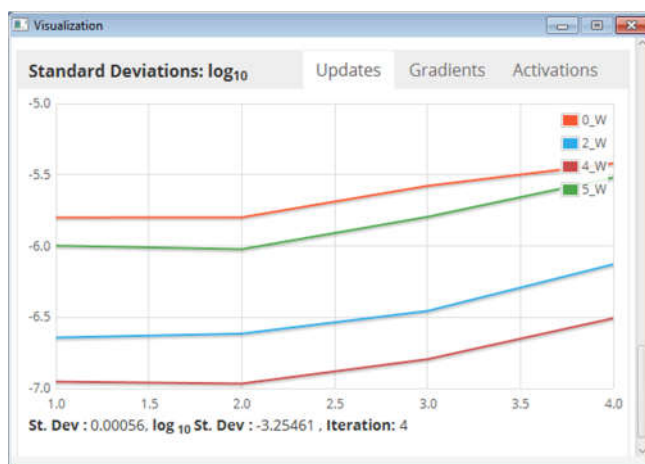


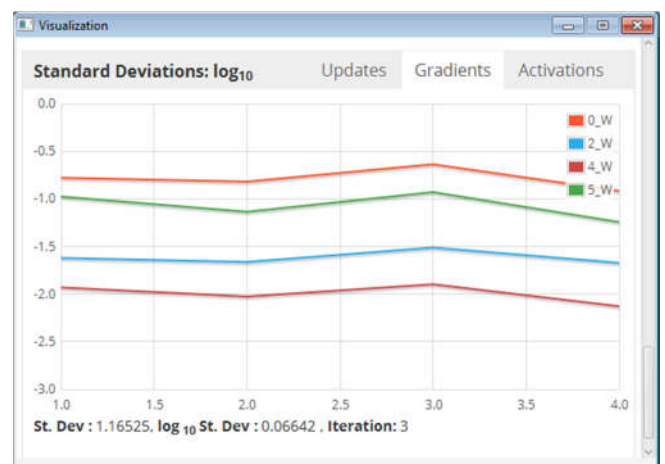
Рисунок 3.15 – Загальна інформація про процес навчання

Як видно з даного рисунку, функція візуалізації надає можливість переглянути такі кількісні показники процесу навчання як: кількість шарів мережі, загальна кількість змінних параметрів, кількість оновлень в секунду, кількість опрацьованих зображень в секунду та ін.

На рисунках 3.16 а), б) та 3.17 а) наведено приклад графіків стандартного відхилення (від часу) оновлень, градієнтів та активацій.



а)



б)

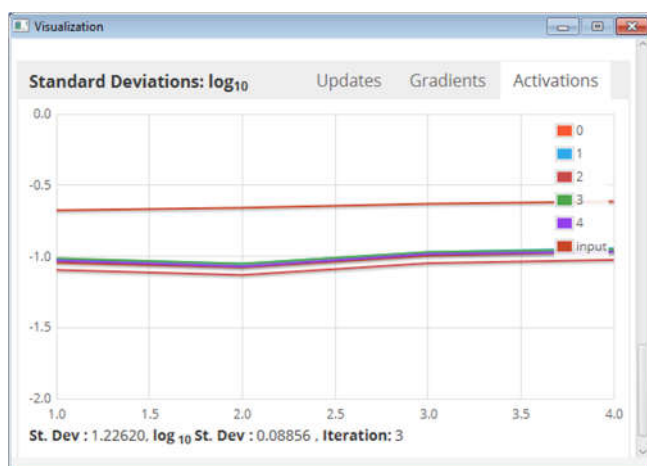
Рисунок 3.16 – а) Стандартне відхилення оновлень моделі;

б) Стандартне відхилення градієнтів моделі

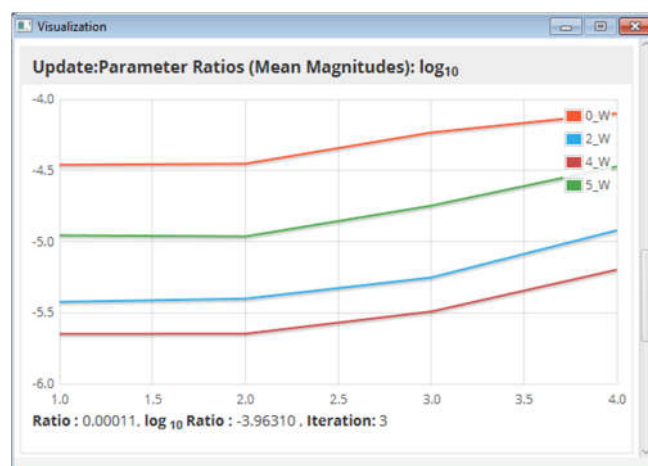
Ці діаграми можуть використовуватися для виявлення зникнення або вибуху активації (через низьку ініціалізацію ваги, надмірну регуляризацію, відсутність нормалізації даних або занадто високу швидкість навчання). Ця діаграма має ідеально стабілізуватися з часом. Добре стандартне відхилення для активації становить від 0,5 до 2,0. Значно за межами цього діапазону може вказуватися одна з проблем, згаданих вище.

Гістограма оновлень шару відображається лише для останньої ітерації. Як і для графіків параметрів, графіки відхилення градієнтів моделі повинні мати приблизно Гаусовий (нормальний) розподіл. Важливо слідкувати за дуже великими значеннями: це може означати вибухові градієнти у мережі. Вибухові градієнти є проблематичними, оскільки вони можуть "зіпсувати" параметри мережі.

На рисунку 3.17 б) зображено приклад графіку зміни середнього абсолютного значення параметрів або оновлень моделі.



а)



б)

Рисунок 3.17 – а) Стандартне відхилення активацій моделі,

б) Середнє абсолютне значення параметрів або оновлень на поточному відрізку часу

Найважливішим використанням цього співвідношення є вибір курсу навчання. Як правило, це співвідношення повинно бути приблизно $1:1000 = 0,001$. На (\log_{10}) діаграмі це відповідає значенню -3 (тобто $10^{-3} = 0,001$)

Якщо співвідношення значно відрізняється від цього (наприклад, > -2 (тобто, $10^{-2} = 0,01$) або < -4 (тобто $10^{-4} = 0,0001$), параметри можуть бути надмірно нестабільними для вивчення корисних функцій або можуть змінюється надто повільно, щоб навчитися корисним функціям. Щоб змінити це співвідношення, необхідно налаштувати швидкість навчання (або іноді ініціалізацію параметра).

На рисунку 3.18 зображено приклад вікна моделі, яка містить граф шарів нейромережі, який працює як механізм вибору. Щоб відобразити детальну інформацію про шар мережі, необхідно натиснути на нього.

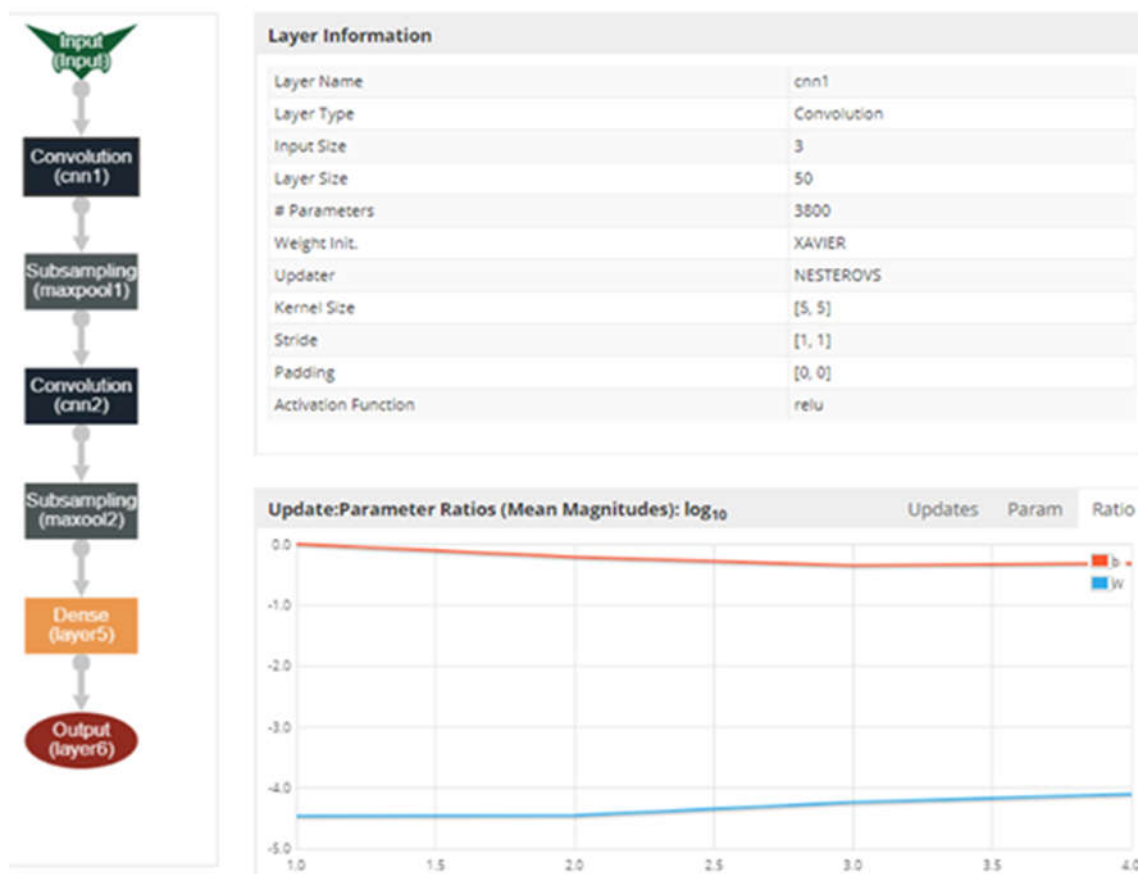


Рис
уно
к
3.18
–
Вік
но з
дет
аль
ною
інф
орм
аціє

ю про модель

Як видно з рисунку 3.18, вікно інформації про модель складається з області, де відображено граф мережі та область з детальною інформацією про вибраний

шар мережі. Сюди входять дані про оновлення шару, зміну параметрів, активації. Для прикладу, можна переглянути швидкість навчання параметрів вибраного шару з часом.

Крім цього, надається можливість переглянути параметри кожного шару. А саме: тип шару, розмір, кількість змінних параметрів, розмір ядра, функцію активації та ін.

На рисунку 3.19 наведено інформацію про використання програмного та апаратного забезпечення.

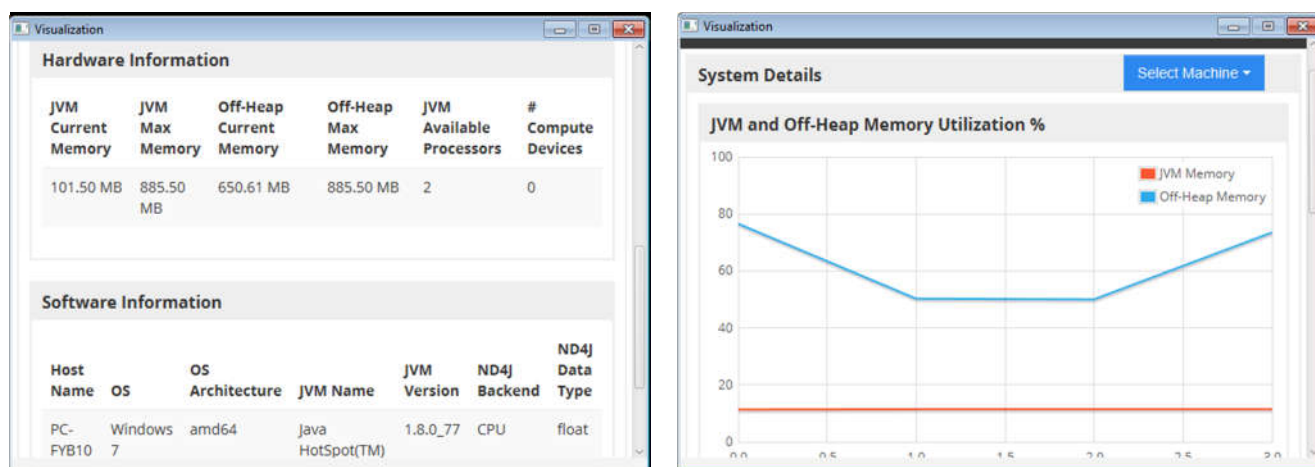


Рисунок 3.19 – Інформація про програмне та апаратне забезпечення

Як видно з рисунку 3.19 функція візуалізації надає можливість переглянути інформацію про використання оперативної пам'яті (поточне та максимальне), процесора (кількість доступних обчислювальних ядер). Крім цього доступною є інформація про використовуване програмне забезпечення (операційна система, архітектура ОС, версія Java Virtual Machine та ін.).

Окрім вищеприведеного функціоналу, в розробленому програмному модулі розпізнавання гістологічних зображень присутня функція тестування якості класифікації попередньо навченою нейронною мережею (рисунок 3.20).

На рисунку 3.20 цифрами позначено:

1. шлях на диску до файлу попередньо навченої мережі;
2. кнопка, яка відкриває діалог вибору шляху до файлу мережі;
3. шлях на диску до зображення, яке потрібно класифікувати;

- кнопка, яка відкриває діалог вибору шляху зображення, клас якого необхідно визначити;
- кнопка, яка запускає процес класифікації вибраного зображення вказаною мережею;
- область результатів класифікації.

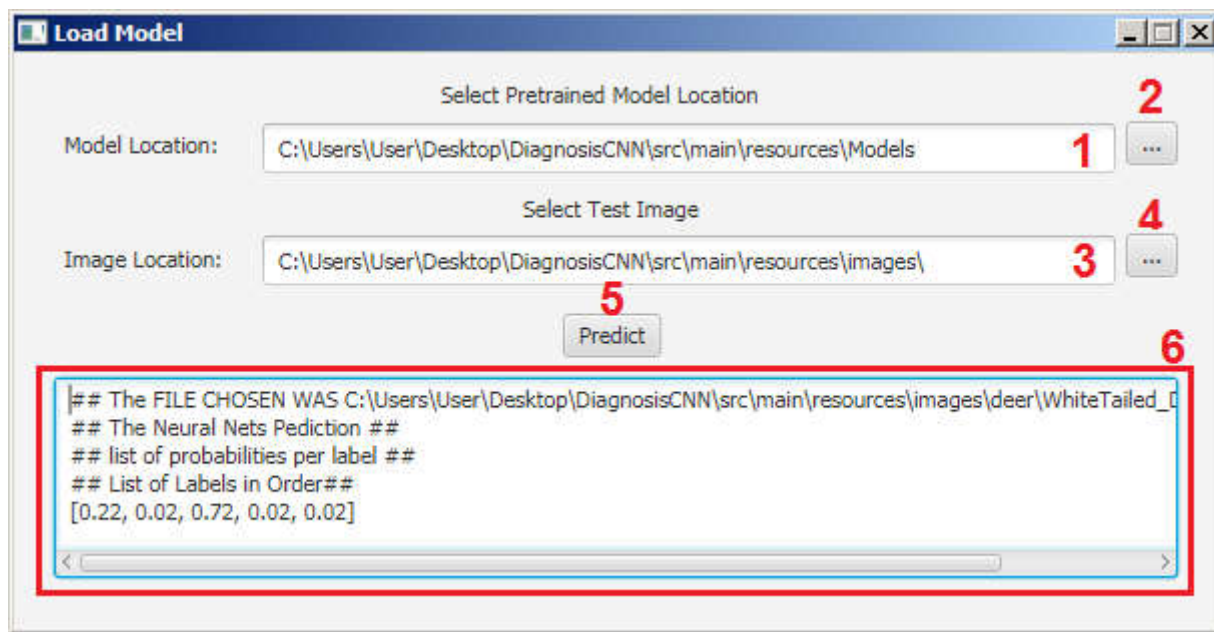


Рисунок 3.20 – Вікно тестування якості класифікації попередньо навченою нейронною мережею

Як видно з рисунку 3.20, результат класифікації для вибраного зображення представлений у вигляді вектора вагових коефіцієнтів. Кожен елемент цього вектора являє собою відповідний клас зображення. Чим більше значення елемента вектора, тим більша імовірність приналежності даного зображення до відповідного класу.

3.4. Тестова вибірка та тестування програмного модуля

Перед тим, як скористатися нейронною мережею, необхідно дослідити ступінь достовірності результатів обчислень мережі на тестовому масиві вхідних зображень. В якості тестового масиву необхідно використовувати ті зображення, які не використовувалися для навчання згорткової нейронної мережі. Важливо перевіряти достовірність роботи мережі саме на тих зображеннях, які не використовувалися для навчання, бо дуже часто трапляються випадки, коли мережа показує хороші результати класифікації на навчальних зображеннях, а на тестовій вибірці – набагато гірші.

Сформуємо тестову вибірку з наступних класів гістологічних зображень патологічних станів молочної залози: непроліферативна мастопатія, проліферативна мастопатія, фіброаденома, внутрішньопротоковий рак, інвазивний рак. На рисунку 3.21 наведено приклади зображень, які увійшли у вибірку.

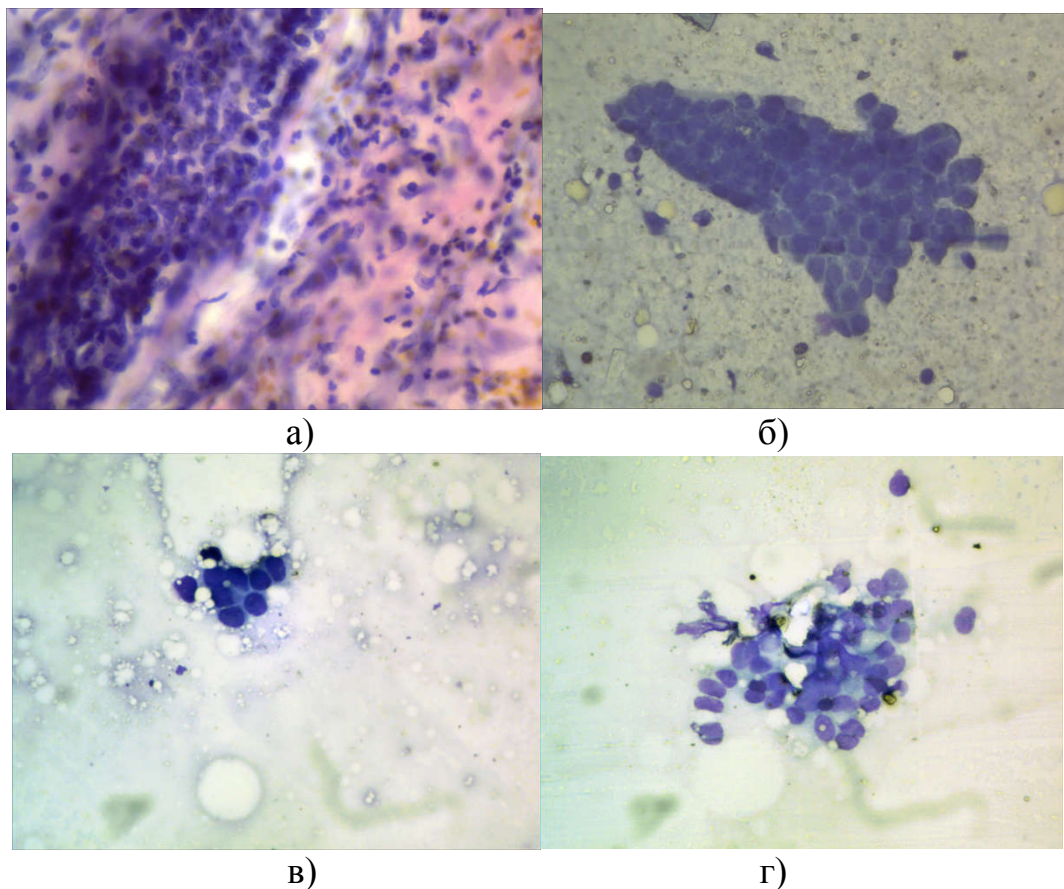


Рисунок 3.21– Приклад зображень, які увійшли в тестову вибірку

Як правило, навчальну і тестову вибірки формують із співвідношенням 70%/30%. Де 70% - це та частина всіх зображень, яка використовується для тренування мережі, а 30% - частина зображень, яка використовується для перевірки достовірності класифікації (тестова вибірка).

Тестування згорткової нейронної мережі проходило на основі бази даних гістологічних зображень. Розроблена БД [53] доповідалася на конкурсі наукових робіт в 2015 році і була відмічена дипломом III ступеня (додаток В). Крім цього, за результатами досліджень з попереднього оброблення гістологічних та цитологічних зображень, було опубліковано тези [54]. А результати дослідження процесу розпаралелення попереднього оброблення опубліковано в [55].

До складу об'єкту дослідження (гістологічні зображення раку молочної залози) входять зображення таких видів раку (класів): непроліферативна мастопатія, проліферативна мастопатія, фіброаденома, внутрішньопротоковий рак, інвазивний рак. Нижче наведений якісний опис кожного з цих класів.

Тестування проводилося на двох вибірках однакових зображень. Різниця у вибірках в тому, що перша вибірка – оригінальні зображення, а друга – зображення, які попередньо пройшли попереднє оброблення.

В якості попереднього оброблення використовувалася медіанна фільтрація на GPU. В даному випадку при розробці методу медіанної фільтрації на GPU було використано фільтрацію методом квадрату.

В результаті проведення попереднього оброблення наведено такі характеристики продуктивності (таблиця 3.1)

Таблиця 3.1 – Характеристики продуктивності обробки зображення

	Час виконання, мс		Прискорення, раз	
	CPU	GPU		
3x3 1920x1080	60	23	2,6	
3x3 4096x2160	140	90	1,55	

Крім цього, було випробувано три різні згорткові нейронні мережі (рисунок 3.22):

- 1) Conv 3x3 – Pool 3x3 – Conv 5x5 – Pool 3x3 – Conv 3x3 – Conv 4x4 – Conv 5x5 – Pool 3x3 – Fully connected;
- 2) Conv 3x3, Pool 3x3, Conv 4x4, Conv 5x5, Conv 6x6, Pool 3x3, Conv 3x3 – FC;
- 3) Conv 5x5, Pool 3x3, Conv 4x4, Pool 3x3, Conv 3x3, FC;

де Conv – згортковий шар, Pool – шар субдискретизації, FC (Fully connected) – повнозв'язний шар.

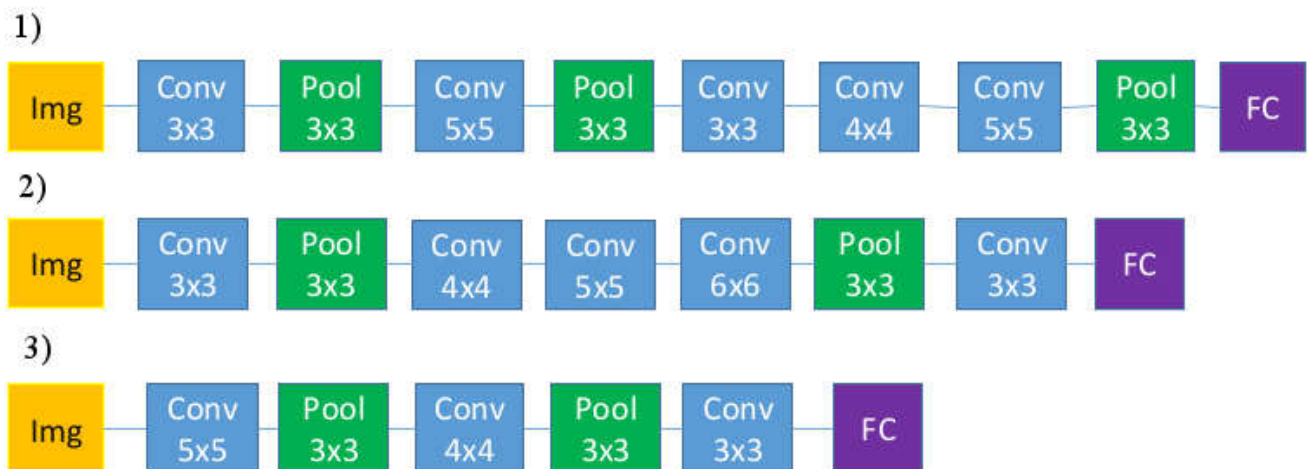


Рисунок 3.22 – Тестові структури згорткових нейронних мереж

В таблиці 3.2 наведено результати тестування (E – кількість епох, N – кількість зображень у навчальній вибірці).

Таблиця 3.2 – Результати тестування класифікації гістологічних зображень

Мережа	Попереднє оброблення	E = 20 N = 20	E = 20 N = 50	E = 50 N = 20	E = 50 N = 50
1	Так	0.4	0.62	0.721	0,8358
	Ні	0.4	0.6	0.70	0.812
2	Так	0.42	0.654	0.732	0.84
	Ні	0.4	0.62	0.726	0.84
3	Так	0.2	0.421	0.52	0.6412
	Ні	0.2	0.4	0.49	0.6382

В таблиці 3.2 наведено значення якості роботи мережі під час тренування (F1 Score). Це число між 0 та 1, яке означає ймовірність того, що наша згортова нейронна мережа дасть правильний результат при певному зображенні на вході. На рисунку 3.23 наведено графічне представлення результатів тестування.

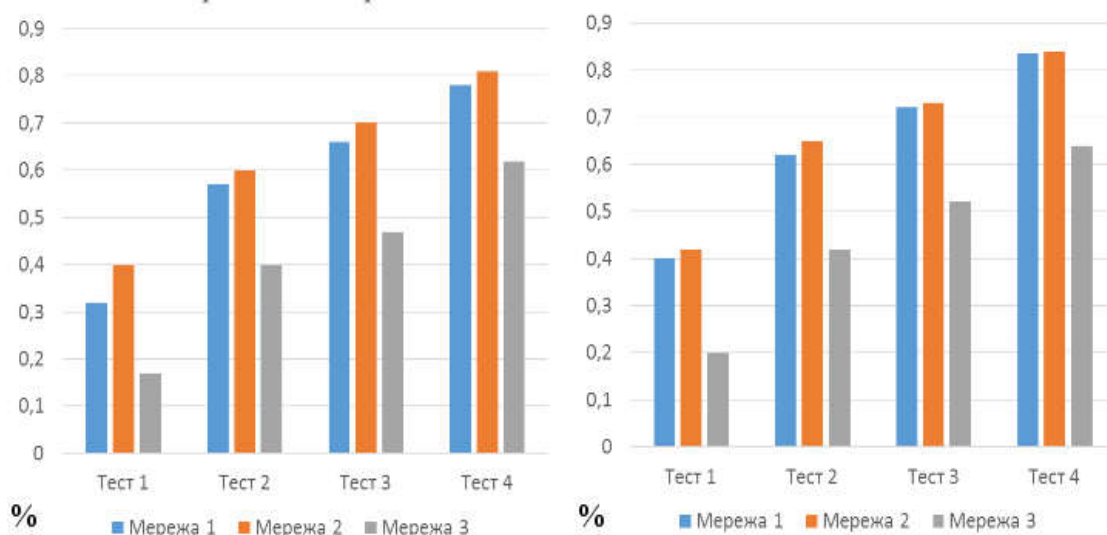


Рисунок 3.23 – Графічне представлення результатів тестування нейронних мереж

Як видно з таблиці 3.2 та рисунку 3.22 – точність роботи згорткової нейронної мережі для класифікації гістологічних зображень раку молочної залози

прямо пропорційно залежить від розміру навчальної вибірки та кількості епох під час навчання.

На рисунках 3.24 та 3.25 наведено діаграми помилок першого та другого роду для трьох згорткових нейронних мереж, на яких проводилось тестування.

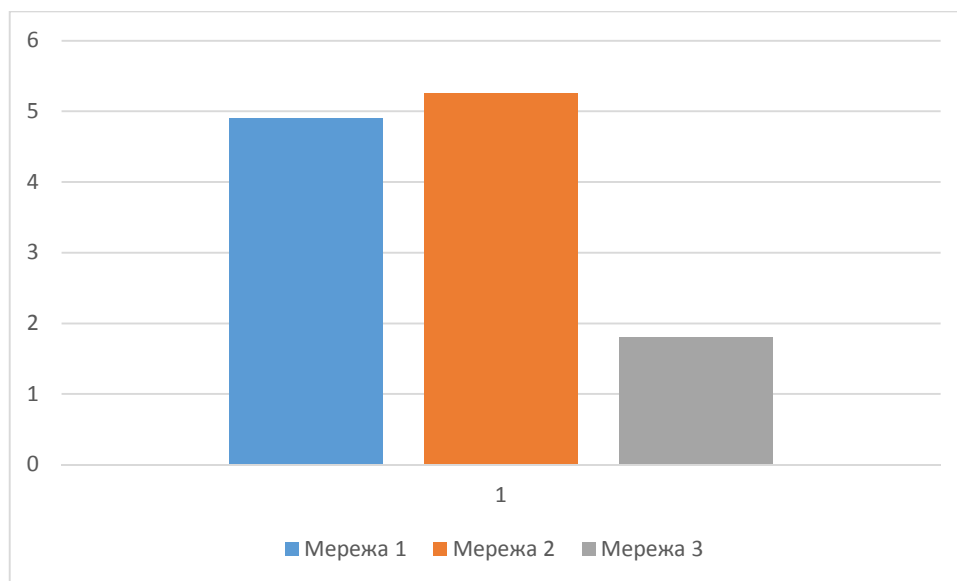


Рисунок 3.24 – Помилки першого роду

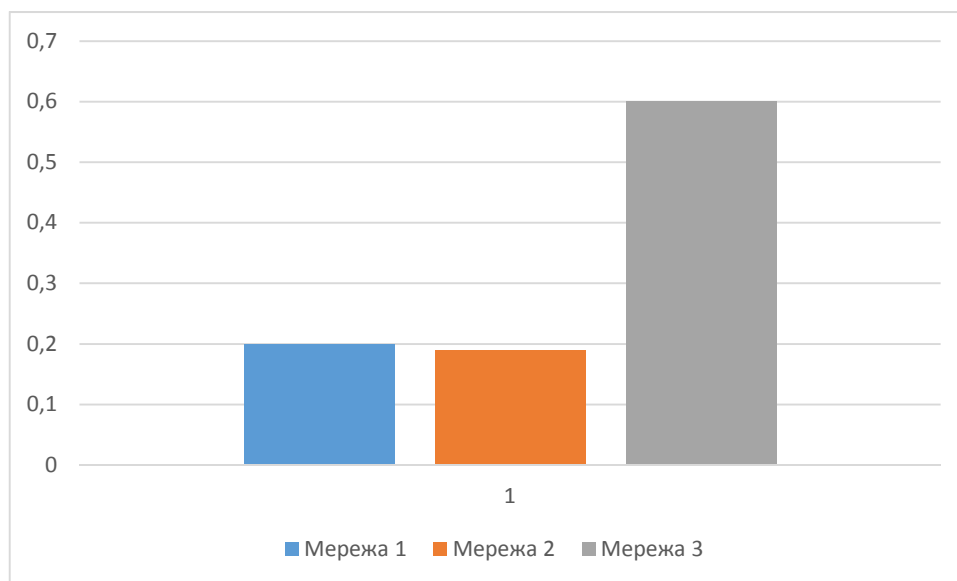


Рисунок 3.25 – Помилки другого роду

Наявність попереднього оброблення тестових зображень також позитивно впливає на якість навчання і, відповідно, на якість і точність класифікації.

Як видно з рисунків 3.24 і 3.25 – згідно з аналізу помилок класифікації першого та другого роду, найкращий результат показала друга (рисунок 3.22) згорткова нейронна мережа.

ВИСНОВКИ

1. Приведено опис і класифікацію гістологічних зображень раку молочної залози, що дало змогу виділити характерні ознаки для кожного класу та для гістологічних зображень загалом.

2. Проведено дослідження згорткових нейронних мереж, їх топології, алгоритми розпізнавання зображень в них та їх навчання, що дало змогу виділити основні ознаки та структуру згорткових нейронних мереж.

3. Розроблено програмний модуль класифікації гістологічних зображень раку молочної залози на основі згорткових нейронних мереж, який надає можливість побудови згорткової нейронної мережі, її навчання та тестування.

4. Проведено тестування розробленого програмного модуля шляхом навчання трьох різних за структурою згорткових нейронних мереж на двох навчальних вибірках: з попереднім обробленням та без. Це дало змогу визначити оптимальну структуру згорткової нейронної мережі, тип та кількість шарів для задачі класифікації гістологічних зображень раку молочної залози.

5. Попереднє оброблення зображень зменшує рівень шумів на зображенні, що зменшує помилки розпізнавання зображень.

6. В результаті тестування було визначено, що якість і точність роботи згорткової нейронної мережі для класифікації гістологічних зображень раку молочної залози прямо пропорційно залежить від величини навчальної вибірки та від кількості епох під час навчання. Чим їх більше, тим кращі результати видає мережа. Наявність попереднього оброблення зображень перед навчанням позитивно впливає на якість навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бюлетень Національного канцер-реєстру № 17 - "Рак в Україні, 2014-2015". - 2016 - 96 с.
2. Бриндіков Л.М. Цитологічна діагностика захворювань молочної залози, органів грудної порожнини та лімфатичних вузлів / Л.М. Бриндіков, А.Д. Давиденко, П.М. Боцюн, І.Г. Дробнер / Хмельницький обласний онкологічний диспансер.
3. Луцик О.Д. Гістологія людини / О.Д. Луцик, А.Й. Іванова, К.С. Кабак, Ю.Б. Чайковський – Київ, «Книга Плюс», 2003. – 593 с.
4. Кількісна гістологія [Електронний ресурс]. – Режим доступу - https://uk.wikipedia.org/wiki/Кількісна_гістологія.
5. Кількісна гістологія [Електронний ресурс]. – Режим доступу - <http://quantitative.histology.org.ua>.
6. Bourzac K. Software: The computer will see you now / K. Bouczak / - Nature, 2013, 502(7473): S92-S94.
7. Гістологічні методи дослідження [Електронний ресурс]. – Режим доступу - <http://ukrbukva.net/page,3,94380-Gistologicheskie-metody-issledovaniya.html>.
8. Предмет гістології [Електронний ресурс]. – Режим доступу- <http://studfiles.net/preview/5258227>.
9. Методы гистологического анализа [Електронний ресурс]. – Режим доступу - <http://worldofscience.ru/biologija/6481-metody-gistologicheskogo-analiza.html>.
10. Автандилов Г.Г. Основы количественной паталлогической анатомии / Г.Г. Автандилов / – М.: Медицина , 2002. – 240 с.
11. Теорія розпізнавання образів [Електронний ресурс]. – Режим доступу - https://uk.wikipedia.org/wiki/Теорія_розпізнавання_образів
12. Вапник В.Н. Теория распознавания образов / В.Н. Вапник, А.Я. Червоненкис. — М.: Наука, 1974. — 416 с.
13. Васильев В.И. Распознающие системы. Справочник. — 2-е изд. / В.И.

Васильев. — К.: Наукова думка, 1983. — 424 с.

14. Линейный классификатор [Электронный ресурс]. – Режим доступа - http://www.machinelearning.ru/wiki/index.php?title=Линейный_классификатор

15. Классификация данных методом опорных векторов [Электронный ресурс]. – Режим доступа - <https://habrahabr.ru/post/105220/>

16. Метод опорних векторів [Электронный ресурс]. – Режим доступа - https://uk.wikipedia.org/wiki/метод_опорних_векторів.

17. Алгоритм AdaBoost [Электронный ресурс]. – Режим доступа - <http://www.machinelearning.ru/wiki/index.php?title=AdaBoost>

18. Топ-10 data mining-алгоритмов простым языком [Электронный ресурс]. – Режим доступа - <https://habrahabr.ru/company/itinvest/blog/262155/>

19. Breiman, Leo (2001). Random Forests. Machine Learning 45 (1). с. 5–32.

20. Каллан Р. Основные концепции нейронных сетей. The Essence of Neural Networks First Edition. — М.: Вильямс, 2001. — 288 с.

21. neuron_seti_2006.djvu

22. Что такое искусственные нейронные сети? [Электронный ресурс]. – Режим доступа - <https://habrahabr.ru/post/134998>.

23. Элементы нейронных сетей [Электронный ресурс]. – Режим доступа - <https://www.intuit.ru/studies/courses/6/6/lecture/178?page=2>

24. Kerlirzin, P. Robustness in multilayer perceptrons / P. Kerlirzin, F. Vallet / Neural Computation, 1993, vol. 5, p. 473 - 482.

25. Mead C.A. Analog VLSI and Neural Systems, Reading / C.A. Mead / - MA: Addison-Wesley, 1989.

26. Black I.B. Information in the Brain: A Molecular Perspective / I.B. Black / - Cambridge, MA: MIT Press, 1991.

27. McCulloch W.S. A logical calculus of the ideas immanent in nervous activity / W.S. McCulloch, W. Pitts / Bulletin of Mathematical Biophysics, 1943.—№ 5.—P.115-133.

28. MATLAB - The Language of Technical Computing [Электронный ресурс]. – Режим доступа: <http://www.mathworks.com/products/matlab>.

29. MATLAB – Wikipedia [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/MATLAB>.
30. Neural Network Toolbox – Математика [Электронный ресурс]. – Режим доступа: <http://matlab.exponenta.ru/neuralnetwork/>
31. Введение в Neural Network Toolbox [Электронный ресурс]. – Режим доступа: <http://matlab.exponenta.ru/neuralnetwork/book1/>
32. Peltarion [Электронный ресурс]. – Режим доступа: <http://peltarion.com/>
33. The Premier Neural Network Software [Электронный ресурс]. – Режим доступа: <http://www.neurosolutions.com/>
34. Deep Learning for Java [Электронный ресурс]. – Режим доступа: <https://deeplearning4j.org/>
35. Convolutional Neural Networks (LeNet) [Электронный ресурс]. – Режим доступа: <http://deeplearning.net/tutorial/lenet.html>
36. van den Oord Deep content-based music recommendation / van den Oord Deep, Aaron; Dieleman, Sander; Schrauwen, B. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger / Curran Associates, Inc. с. 2643–2651.
37. Korekado A Convolutional Neural Network VLSI for Image Recognition Using Merged/Mixed Analog-Digital Architecture. Knowledge-Based Intelligent Information and Engineering Systems. с. 169–176. CiteSeerX: 10.1.1.125.3812.
38. Krizhevsky Alex. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, E. Hinton / Advances in Neural Information Processing Systems, 2012. - №25. – P.120 – 129.
39. Применение нейросетей в распознавании изображений [Электронный ресурс]. – Режим доступа: <https://geektimes.ru/post/74326/>
40. Классификатор изображений на основе свёрточной сети [Электронный ресурс]. – Режим доступа: <http://mechanoid.kiev.ua/ml-lenet.html>
41. Convolutional Neural Networks (CNNs / ConvNets) [Электронный ресурс]. – Режим доступа: <http://cs231n.github.io/convolutional-networks>
42. Y. LeCun Efficient BackProp / Y. LeCun, L. Bottou, G. Orr and K. Muller / Orr, G. and Muller K. (Eds), Neural Networks: Tricks of the trade, Springer, 1998

43. Алгоритм обратного распространения ошибки [Электронный ресурс]. – Режим доступа: <http://www.aiportal.ru/articles/neural-networks/back-propagation.html>
44. Deep learning for complete beginners: convolutional neural networks with keras [Электронный ресурс]. – Режим доступа: <https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html>
45. The 9 Deep Learning Papers You Need To Know About [Электронный ресурс]. – Режим доступа: <https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>
46. Matthew D. Zeiler Visualizing and Understanding Convolutional Networks / D. Zeiler, R. Fergus / arXiv:1311.2901v3. – 2013
47. Christian Szegedy Going Deeper with Convolutions / Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich / THE COMPUTER VISION FOUNDATION. – 2015, 10p.
48. Ross Girshick Rich feature hierarchies for accurate object detection and semantic segmentation / Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik / arXiv:1311.2524v5. – 2014, 21p.
49. Ross Girshick Fast R-CNN / Ross Girshick // arXiv:1504.08083v2. - 2015, 9p.
50. Shaoqing Ren Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun // arXiv:1506.01497v3 [cs.CV]. – 2016, 14p.
51. Обучение нейронной сети [Электронный ресурс]. – Режим доступа: <http://neuronus.com/theory/240-algoritmy-obucheniya-iskusstvennykh-nejronnykh-setej>
52. Алгоритм обучения многослойной нейронной сети методом обратного распространения ошибки (Backpropagation) [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/198268>.
53. Березький О.М. База даних цитологічних та гістологічних зображень ауто- та ксеногенних тканин / Березький О.М., Мельник Г.М., Дацко Т.В., Вербовий С.О. // Науковий вісник національного лісотехнічного університету України:

збірник науково-технічних праць. – Львів: РВВ НЛТУ України. – 2014.– Вип. 24.10. – С.338-345.

54. Боднар А.Р. Алгоритми попереднього оброблення біомедичних зображень на базі бібліотеки OpenCV / А.Р. Боднар, О.Й. Піцун // Сучасні комп'ютерні інформаційні технології: Матеріали VI Всеукраїнської школи-семінару молодих вчених і студентів АСІТ'2016. – Тернопіль: ТНЕУ, 2016 – с.81-83.

55. Цмоць І.Г. Паралельні алгоритми та нвіс-структури для медіанної фільтрації зображень в реальному часі / І.Г.Цмоць, І.В.Ігнатєв, А.Р. Боднар // Збірник наукових праць Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова. – Київ. – Прийнято до друку.