

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Приладобудівний факультет

Кафедра приладів та систем неруйнівного контролю

«До захисту допущено»

Завідувач кафедри

_____ Протасов А.Г.

«__» _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

з напрямку підготовки 6.051003 «Приладобудування»

**на тему: «Онлайн-сервіс для діагностування захворювань легень за
рентгенівськими знімками»**

Виконав (-ла):

студент (-ка) IV курсу, групи ПК-51

Мерецький Микола Михайлович

Керівник:

к.т.н доцент

Галаган Р.М.

Рецензент:

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент (-ка) _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Приладобудівний факультет

Кафедра приладів та систем неруйнівного контролю

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.051003
«Приладобудування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Протасов А.Г.

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Мерецький Микола Михайлович

1. Тема роботи «Онлайн-сервіс для діагностування захворювань легень за рентгенівськими знімками», керівник роботи Галаган Роман Михайлович, к.т.н, затверджені наказом по університету від «27» травня 2019 р. № 1384-с
2. Термін подання студентом роботи 10 травня 2019
3. Вихідні дані до роботи: онлайн-сервіс; нейронна мережа для обробки знімків; тип нейронної мережі – Sequential; кількість шарів - 19;
4. Зміст пояснювальної записки

Вступ

1. Аналітичний огляд
2. Огляд машинного навчання
3. Розробка програмного забезпечення
4. Результати дослідження

Висновок

Список використаних джерел

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) 4 плакати

1 – Загальні положення про рентгенологію

2 – Загальні положення про пневмонію

3 – Огляд машинного навчання

4 – Онлайн-сервіс для діагностування захворювань легень за рентгенівськими знімками

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
-	-		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Формулювання завдання проекту	15.03.2019	
2	Проведення аналітичного огляду	10.04.2019	
3	Дизайн архітектури нейронної мережі	25.04.2019	
4	Дизайн серверної архітектури	15.05.2019	
5	Розробка плакатів	12.06.2019	

Студент

Мерецький М.М.

Керівник проекту

Галаган Р.М.

**Пояснювальна записка
до дипломної роботи
на тему: «Онлайн-сервіс для діагностування
захворювань легень за рентгенівськими знімками»**

Київ – 2019 року

Анотація

Дана дипломна робота складається з 66 сторінок, 17 ілюстрацій, 12 формул, 9 літературних джерел.

Ключові слова: нейронна мережа, онлайн-сервіс, python, пневмонія, рентген.

В дипломному проекті було проведено дослідження нового методу діагностики захворювання легень на основі рентген знімків.

Розроблено дизайн архітектури нейронної мережі. Модель було навчено розпізнавати вірусну та бактеріальну пневмонію на рентгенівських знімках. Було розроблено онлайн-сервіс, який, в режимі реального часу, надає зручний та простий доступ до можливостей нейронної мережі.

В результаті дослідження було розроблено готове рішення для сучасної діагностики на основі сучасних нейромережових технологій.

Мета роботи полягає в дослідженні високоточного методу діагностики захворювань легень на основі рентген знімків. Його розробка, розробка онлайн інтерфейсу для зручного доступу в режимі реального часу та складання документації до нього

Об'єкт дослідження - грудна клітка людини, зокрема легені

Предмет дослідження – методи діагностики на основі неромережових технологій

Наукова новизна – новітній метод діагностики захворювання легень на основі рентген знімків.

Abstract

This thesis consists of 66 pages, 17 illustrations, 12 formulas, 9 literary sources.

Key words: neural network, online service, python, pneumonia, x-ray.

In the diploma project, a new method for diagnosing lung disease on the basis of X-ray images was conducted.

The design of the neural network architecture is developed. The model was trained to recognize viral and bacterial pneumonia in X-rays. An online service was developed that provides real-time, convenient and easy access to the capabilities of the neural network.

As a result of the study, a ready-made solution for modern diagnostics was developed on the basis of latest neural network technology.

The **purpose of the work** is to investigate the high-precision method of diagnosing lung diseases on the basis of X-ray images. Also, to development an online interface for easy access in real time and documentation for it

The **object of the study** is the human chest, in particular the lungs

Subject of research - methods of diagnostics based on non-network technologies

Scientific novelty - the newest diagnostic method of lung disease on the basis of X-ray images.

Зміст

СКРОЧЕННЯ ТА ПОЗНАЧЕННЯ	8
ВСТУП	9
1. АНАЛІТИЧНИЙ ОГЛЯД	11
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ ПРО РЕНТГЕНОСКОПІЮ	11
1.2 РЕНТГЕН ГРУДНОЇ КЛІТКИ	14
1.3 ПНЕВМОНІЯ	19
1.3.1 Симптоми пневмонії.....	20
1.3.2 Фактори ризику	23
1.3.3 Ускладнення	23
1.3.4 Профілактика.....	24
1.4 ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ ДЛЯ ДІАГНОСТУВАННЯ ПНЕВМОНІЇ.....	25
1.4.1 Глибинне навчання проти «традиційного» машинного навчання.....	26
1.4.2 Глибинне машинне навчання.....	27
2. ОГЛЯД МАШИННОГО НАВЧАННЯ	29
2.1 ВИЗНАЧЕННЯ НЕЙРОННОЇ МЕРЕЖІ	29
2.2 КЛАСИФІКАЦІЯ	30
2.3 КЛАСТЕРИЗАЦІЯ.....	30
2.4 РЕГРЕСІЙНИЙ АНАЛІЗ.....	31
2.5 ЕЛЕМЕНТИ НЕЙРОННОЇ МЕРЕЖІ	32
2.6 КЛЮЧОВІ КОНЦЕПЦІЇ НЕЙРОННИХ МЕРЕЖ.....	33
2.7 МЕРЕЖІ "FEEDFORWARD"	36
2.8 МНОЖИННА ЛІНІЙНА РЕГРЕСІЯ.....	38
2.9 ГРАДІЄНТНИЙ СПУСК.....	40
2.10 ФУНКЦІЇ АКТИВАЦІЇ.....	41
2.11 ЛОГІСТИЧНА РЕГРЕСІЯ	42
2.12 НЕЙРОННІ МЕРЕЖІ ТА ШТУЧНИЙ ІНТЕЛЕКТ	43
2.13 АНАЛІЗ РІЗНИХ МОВ ПРОГРАМУВАННЯ ДЛЯ РЕАЛІЗАЦІЇ НЕЙРОННИХ МЕРЕЖ	44
2.13.1 Python	44
2.13.2 R.....	44
2.13.3 Scala	45
2.13.4 Julia	45
2.13.5 Java.....	46
2.14 РУТНОН ЯК НАЙКРАЩА МОВА ДЛЯ РОЗРОБКИ ШТУЧНОГО ІНТЕЛЕКТУ	46
2.15 ОСНОВНІ БІБЛІОТЕКИ ДЛЯ МАШИННОГО НАВЧАННЯ В РУТНОН	50
2.15.1 Scikit-learn для роботи з класичними алгоритмами ML	50

2.15.2 Tensorflow для глибинного навчання.....	51
2.15.3 Theano для Deep Learning.....	51
2.15.4 Pandas для вилучення та підготовки даних.....	52
2.15.4 Matplotlib для візуалізації даних.....	52
3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	54
3.1 ВИБІР МОВИ ПРОГРАМУВАННЯ.....	54
3.2 ВИБІР БІБЛІОТЕК ДЛЯ PУТНОН.....	55
3.3 ПІДГОТОВКА ДАТАСЕТУ.....	55
3.4 ДИЗАЙН АРХІТЕКТУРИ МОДЕЛІ.....	58
3.5 НАВЧАННЯ МОДЕЛІ.....	59
3.6 ТЕСТУВАННЯ МОДЕЛІ.....	60
3.7 РОЗРОБКА ОНЛАЙН СЕРВІСУ.....	62
4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ.....	63
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	67

Скорочення та позначення

AI – artificial intellect

CRM – customer relationship management

DL – deep learning

ML – machine learning

КТ – комп'ютерна томографія

МРТ – магнітно-резонансна томографія

МН – машинне навчання

НМ – нейронна мережа

ШІ – штучний інтелект

Вступ

В теперішньому світі медицина відіграє надзвичайно важливу роль для життя та існування кожної людини. Внаслідок постійного розвитку технологій, постійного забруднення довколишнього середовища, зростає кількість хворіб та хворих людей. Саме по цій причині необхідно підтримувати розвиток медицини та постійно вдосконалювати існуючу систему, застосовуючи нові технології та методи.

В діагностичних цілях широко використовуються методи інтроскопічного дослідження, які об'єднані загальним терміном – променева діагностика. В даній групі знаходяться всі види традиційного рентгенологічного дослідження – рентгеноскопія, рентгенографія, лінійна томографія та ін., а також інші сучасні методи, такі як рентгенівська геліолокальна і спіральна комп'ютерна томографія і методи радіонуклідної діагностики.

Кожен з наведених вище методів має певний ряд переваг та недоліків і всі вони відрізняються певними межами діагностичних можливостей і сферами застосування в медицині. В системі клінічного та профілактичного дослідження населення, методи променевої діагностики займають одне з передових місць в системі клінічного та профілактичного дослідження населення. З їхньою допомогою ставиться приблизно 60-80% всіх первинних діагнозів. Найбільш поширеним методом променевого дослідження являється рентгенографія.

Завдяки введенню в практичну сферу охорони здоров'я новітніх комп'ютеризованих технологій, сучасних програмних засобів, способів цифрової обробки зображень, можливості і роль методів променевої діагностики в медицині ще більше зростають. Постановка діагнозу, яка здійснюється за рентгенівськими знімками вже дуже давно відома медицині, але цей метод діагностування можна зробити набагато ефективнішим. Для цього потрібно частково виключити похибки, спричинені людьми, завдяки використанню в рентгенології наперед навчених нейронних мереж. Це може допомогти роботі постановку діагнозу куди швидшою та точнішою, адже

частково позбуваючись людського фактору, ми також частково позбуваємось достатньої кількості помилок, що були спричинені людьми.

Нейронні мережі – це один з найбільш популярних напрямків наукових досліджень в галузі створення штучного інтелекту, в основі якого лежить прагнення імітувати нервову систему людини, в тому числі її здатність виправляти помилки і вчитись на пройдених помилках. Все це, хоча і дещо узагальнено повинно дозволити змодельовати роботу людського мозку. Нейронна мережа це не тільки математична модель, а також її апаратна чи програмна реалізація. Після розробки алгоритму навчання, отримується модель з певними заданими вхідними та вихідними даними, яка може використовуватись в різних практичних цілях: в задачах прогнозування, для розпізнавання образів, задачах управління та інших [1, 2].

В даній роботі буде розглянуто онлайн-сервіс, на основі нейронної мережі, який буде допомагати лікарям при постановці діагнозу пневмонії. Даний сервіс має ряд вагомих переваг над звичними методами діагностики:

- висока точність
- постійний доступ в режимі онлайн
- відсутність необхідності в медичній освіті для проведення діагностики
- відсутність людського фактору, і, як наслідок, відсутність помилок через неуважність, тощо.
- короткий час поставлення діагнозу (обмежується швидкістю інтернет з'єднання)
- висока продуктивність

1. Аналітичний огляд

1.1 Загальні положення про рентгеноскопію

Рентгенівський апарат виробляє контрольований промінь випромінювання, який використовується для створення зображення вашого тіла. Цей промінь спрямований на досліджувану область. Пройшовши крізь тіло, промінь падає на плівку або спеціальну пластину, де він відкидає певну тінь. Різні тканини в тілі блокують або поглинають випромінювання по-різному. Щільна тканина, така як кістка, блокує більшу частину випромінювання і виглядає білою на плівці. М'які тканини, такі як м'язи, блокують менше випромінювання і виглядають темніше на плівці. Часто декілька зображень беруться з різних ракурсів, тому доступний повний перегляд області. Зображення, отримані під час рентгенівських обстежень, можуть бути переглянуті на плівці або після процесу, який називається «оцифрування», на екрані комп'ютера.

Іноді рентгенологічне обстеження включає контраст. Для контрастного дослідження ви приймаєте препарат, який називається контрастним засобом, який підкреслює або контрастує частини тіла, щоб вони проявлялись більш чітко на рентгенівському знімку.

Рентгенівські обстеження можна використовувати для перегляду, моніторингу або діагностики таких хворіб:

- переломи кісток
- пошкодження суглобів та інфекцій
- блокування артерій
- рак
- пневмонії

Різні типи рентгенівських променів [3]:

- Рентген грудної клітки - рентген грудної клітки зображує серце, легені, кровоносні судини, дихальні шляхи і лімфатичні вузли у

цій області. Грудна клітка, ребра і ключиця також включені в зображення. Цей рентгенівський знімок також частково показує верхню частину хребта. Рентген грудної клітки використовується для діагностики проблем зі здоров'ям, які викликають симптоми в області грудей, такі як легеневі та дихальні проблеми та проблеми з серцем. У більшості випадків рентген грудної клітки включає в себе два зображення, один знімається зі спини, а інший береться збоку.

- Хребетний рентген - хребетний рентген зображує хребет і допомагає виявити хвороби і травми, що впливають на хребет, суглоби і диски. Деякі приклади проблем зі здоров'ям з діагнозом хребта - переломи хребта, вивихи, проблеми з дисками, інфекції та пухлини серед кількох хребців. Проте рентгенівські промені хребта найчастіше використовуються при діагностиці проблем, пов'язаних з хребтом, таких як сколіоз або Spina Bifida. Існують різні підтипи хребетних рентгенівських променів залежно від конкретної частини хребта. Пацієнти можуть потребувати шийного, грудного, крижового або куприка, або попереково-крижового рентгену.

- Рентгенографія кінцівок - рентгенографія кінцівок відноситься до рентгенівських сканів кінцівок, а саме руки, зап'ястя, стопи, коліна, ніг, щиколоток і стегон. Це зазвичай необхідне для виявлення та обстеження травм і захворювань, таких як переломи і вивихи. Деякі приклади умов, які впливають на кінцівки і можуть вимагати такого рентгенограми, включають пухлини, остеопороз і артрит.

Для більшості рентгенівських обстежень немає спеціальної підготовки. Вас попросять носити лікарняну сукню і видалити всі прикраси і металеві предмети перед тестом.

Для контрастних рентгенологічних досліджень вам буде надана доза контрастної речовини або орально, або як клізма, або як ін'єкція або катетер (тонка трубка) в певну ділянку тіла. Ваш лікар буде надавати всі конкретні інструкції, необхідні для вашого контрастного дослідження.

Вас попросять або лежати або стояти поруч з рентгенівським апаратом під час дослідження. Кімната може бути прохолодною, щоб пристрій не перегрівався. Технолог або особа, яка виконує іспит, може використовувати подушки або мішки з піском, щоб допомогти вам утримувати належне положення. Вас попросять тримати дуже тихо, без дихання протягом декількох секунд. Технолог піде за радіаційним бар'єром і активує рентгенівський апарат. Часто кілька зображень або переглядів беруться з різних ракурсів, тому технолог пересуне вас в інше положення і процес буде повторений. Ви не відчуєте випромінювання.

Через рентгенологічне обстеження пацієнти отримують певну дозу випромінювання. Кількість радіаційного впливу змінюється залежно від рентгенівського типу (наприклад, головного мозку, легень або живота) і типу рентгенівського апарату (наприклад, різних моделей і виробників). Оскільки радіаційний вплив є змінним, ризики також є змінними [4]. Будь ласка, зверніться до свого рентгенолога або вашого лікаря, який проводить вам рентгенологічне обстеження, для отримання детальної інформації про радіаційний вплив та можливі ризики.

- Жінкам слід повідомити свого лікаря, що вони є або можуть бути вагітними або на грудному годуванні до будь-якого рентгенологічного зображення. Ваш лікар може порекомендувати інший тип тесту для зниження ризику опромінення дитини.
- Існує рідкісний ризик основної алергічної реакції на контрастну речовину.

Рентгенівські промені записуються на плівку або записуються в цифровому форматі. Рентгенолог, який є лікарем, який спеціалізується на рентгенівських та інших тестах, аналізуватиме та інтерпретуватиме результати вашого рентгенівського дослідження, а потім надсилати звіт своєму особистому лікарю. Для неаварійних ситуацій зазвичай потрібно один день, щоб

інтерпретувати, повідомити і доставити результати. Зверніться до свого персонального лікаря для отримання інформації про результати вашого іспиту.

1.2 Рентген грудної клітки

Рентген грудної клітки - це зображення грудної клітки, де показано ваше серце, легені, дихальні шляхи, судини і лімфатичні вузли. Рентген грудної клітки також показує кістки хребта і грудної клітки, включаючи грудину, ребра, ключицю і верхню частину хребта. Рентгенографія органів грудної клітки є найпоширенішим тестом або рентгенівським зображенням, що використовується для виявлення проблем у грудях.

Рентген грудної клітки може допомогти знайти деякі проблеми з органами та структурами в грудях. Зазвичай знімаються дві картини, одна з задньої частини грудної клітки, а інша - з боку. У надзвичайній ситуації, коли знімається тільки одна рентгенівська картина, зазвичай робиться вигляд спереду. Лікарі можуть не завжди отримувати необхідну їм інформацію від рентгенівського знімка грудної клітки, щоб знайти причину проблеми. Якщо результати рентгенографії грудної клітки не є нормальними або не дають достатньої інформації про проблему грудної клітки, можуть бути зроблені більш специфічні рентгенівські знімки або інші тести, такі як комп'ютерна томографія (КТ), ультразвук, ехокардіограма або МРТ.

Рентген грудної клітки виконується для:

- Пошуку причин поширених симптомів, таких як кашель, задишка або біль у грудях.
- Пошуку хворіб легень, таких як пневмонія, рак легень, хронічна обструктивна хвороба легень (ХОЗЛ), колапс легень (пневмоторакс), або кістозний фіброз, і контролювання лікування цих захворювань.
- Пошуку деяких проблем з серцем, таких як збільшене серце, серцева недостатність і проблеми, що викликають рідину в легенях (набряк легень), і контролювання лікування цих захворювань.

- Слідкування за проблемами, пов'язаними з травмою грудей, наприклад, переломами ребер або пошкодженням легень.
- Пошуку сторонніх предметів, таких як монети або інші дрібні шматки металу, у трубці до шлунку (стравоходу), у дихальних шляхах або легенях. Рентген грудної клітки може не бачити їжі, горіхів або деревних волокон.
- Переконавання, що трубка, катетер або інший медичний прилад поміщений у належне положення в дихальних шляхах, серці, кровоносних судинах грудної клітки або шлунку.

Рентген грудної клітки зазвичай не робиться під час вагітності, оскільки радіація може завдати шкоди майбутній дитині (плоду). Але ймовірність шкоди плоду дуже мала. Якщо потрібен рентген грудної клітки, зазвичай одягають фартух, щоб захистити дитину.

Рентген грудної клітки приймається технологом радіології. Зображення зазвичай читає рентгенолог, який пише звіт. Інші види лікарів, такі як сімейний лікар, терапевт або хірург, також можуть переглядати рентген грудної клітки.

Під час дослідження потрібно зняти прикраси, які можуть бути на шляху рентгенівського знімка. Може знадобитися зняти весь або більшу частину одягу над талією (може бути дозволено залишити нижню білизну, якщо вона не перешкоджає тесту). Зазвичай видають плаття для носіння під час дослідження.

Як правило, робляться два види рентгенівських знімків грудей. Один знімок робиться зі спини; інший – збоку тіла. Але деколи можуть знадобитися інші види знімків в залежності від того, що шукає лікар. У надзвичайній ситуації може бути зроблено лише одне зображення, як правило, спереду.

Зазвичай потрібно стояти фронтом на рентгенівській пластині для знімків. Якщо вид рентгенівського знімку потребує сидяче або лежаче положення, то хтось допоможе пацієнту потрапити в правильне положення.

Під час рентгенівського знімання потрібно мінімізувати рухи тіла, щоб запобігти розмиванню зображення. Пацієнта можуть попросити затримати дихання на кілька секунд під час зйомки рентгенівського знімка.

Більшість лікарень і деякі клініки мають портативні рентгенівські апарати. Якщо рентген грудної клітки виконується портативним рентгенівським апаратом біля ліжка в лікарні, рентгенівський технолог і медсестра допоможуть пацієнту перейти в правильне положення. Зазвичай знімається лише одна картинка з фронту.

Під час рентгенографії болю нема. Рентгенівська пластина може відчуватись твердою, і кімната може бути прохолодною. Якщо пацієнт відчуває біль від проблеми у грудях, то може з'являтися певний дискомфорт під час якщо вам потрібно зайняти певну позицію, глибоко вдихнути або затримати дихання під час рентгенографії.

Завжди існує невелика ймовірність пошкодження клітин або тканин від радіації, включаючи низькі рівні радіації, що використовуються для цього випробування. Але ймовірність пошкодження від рентгенівських променів зазвичай дуже низька в порівнянні з перевагами тесту.

Рентген грудної клітки - це зображення грудей, щоб побачити ваше серце, легені, дихальні шляхи, кровоносні судини і лімфатичні вузли. Рентген грудної клітки також показує кістки, включаючи грудину, ребра, ключицю і верхню частину хребта.

У надзвичайній ситуації результати рентгенологічного дослідження можуть бути доступні протягом декількох хвилин для огляду лікарем. Якщо це не надзвичайна ситуація, результати зазвичай готуються через 1 або 2 дні.

Табл. 1.1. Основні ознаки здоров'я органів грудної клітки

Нормально	Легені виглядають нормальними за розміром і формою, а легенева тканина виглядає нормальною. Ніякі нарости або інші маси не видно в легенях. Плевральні простори (простори, що оточують легені) також виглядають нормальними.
	Серце виглядає нормальним за розміром, формою, а тканина серця виглядає нормальною. Кровоносні судини, що ведуть до і від серця, також є нормальними за розміром, формою і зовнішнім виглядом.
	Кістки, включаючи хребет і ребра, виглядають нормально.
	Діафрагма виглядає нормальною формою та розташуванням.
	Не спостерігається аномального збору рідини або повітря, і ніяких сторонніх предметів не видно.
	Всі трубки, катетери або інші медичні пристрої знаходяться у правильному положенні в грудях.
	Існує інфекція, наприклад, пневмонія або туберкульоз.
	Проблеми, такі як пухлина,

Неенормально	<p>травма або стан, наприклад набряк від серцевої недостатності, можуть бути помічені. У деяких випадках може знадобитися більше рентгенівських променів або інших тестів, щоб чітко побачити проблему.</p>
	<p>Виявляється проблема, наприклад, розширене серце, яке може бути викликане ураженням серця, захворюванням серцевого клапана або рідиною навколо серця. Або проблема кровоносних судин, таких як збільшена аорта, аневризма, або зміцнення артерій (атеросклероз).</p>
	<p>Рідина спостерігається в легенях (набряк легень) або навколо легень (плевральний випіт), або повітря спостерігається в просторах навколо легкого (пневмоторакс).</p>
	<p>Розбиті кістки (переломи) спостерігаються в грудній клітці, ключиці, плечі або хребті.</p>
	<p>Видно збільшені лімфатичні вузли.</p>
	<p>У стравоході, дихальних трубах або легких спостерігається сторонній предмет.</p>

Причини, чому результати тестування не несуть змістовної інформації, включають:

- Пацієнт не в змозі залишитися на місці і затримати дихання, коли його запитують під час тесту.
- Металеві предмети (такі як спинний фіксатор і металеві прикраси або ремені) перешкоджають рентгенівському знімку.
- Ожиріння, яке може приховувати проблеми грудей або ускладнювати отримання рентгенівського зображення.
- Біль у грудях, що може ускладнити глибокий вдих під час рентгенографії.
- Рубці від минулої операції на легенях або зміни від хронічного захворювання, які можуть зробити рентгенівські знімки важкими для читання.
- Зображення, зроблене за допомогою портативної машини, може бути не таким яким, як зображення, зроблене стаціонарною машиною.
- Результати рентгенівського тесту можуть відрізнятися від попередніх результатів тестування, тому що ви пройшли тестування в іншому медичному центрі або у вас був інший тест.
- Якщо рентген грудної клітки є ненормальним, можуть бути зроблені більш специфічні рентгенівські знімки або інші тести, такі як КТ, ультразвук, МРТ або біопсія.
- Деякі умови можуть не з'являтися на рентгенограмі грудної клітки, наприклад, невеликий рак, легенева ебола або інша проблема, яка прихована звичайними структурами в грудях.
- Деякі працівники, такі як люди, які працюють з азбестом, можуть потребувати регулярних рентгенівських знімків грудей для перевірки проблем, викликаних азбестом.

1.3 Пневмонія

Пневмонія - це інфекція, яка запалює повітряні мішки в одному або обох легенях. Повітряні мішечки можуть наповнюватися рідиною або гноєм (гнійним матеріалом), викликаючи кашель з мокротою або гноєм, лихоманкою, ознобом і утрудненим диханням. Різноманітні організми, включаючи бактерії, віруси і гриби, можуть викликати пневмонію.

Пневмонія може бути різною за важкістю: від легкої до небезпечної для життя. Найбільш серйозною вона є для немовлят і дітей молодшого віку, людей старше 65 років, людей із проблемами зі здоров'ям або з ослабленою імунною системою.

1.3.1 Симптоми пневмонії

Ознаки і симптоми пневмонії теж варіюються від легких до важких, залежно від таких факторів, як тип зародка, що викликає інфекцію, ваш вік і загальний стан здоров'я. М'які ознаки і симптоми часто схожі з симптомами застуди або грипу, але вони тривають довше.

Ознаки і симптоми пневмонії можуть включати:

- Біль у грудях при диханні або кашлі
- Плутанина або зміни у розумовому розумінні (у дорослих у віці 65 років і старше)
- Кашель, який може виробляти мокротиння
- Втома
- Лихоманка, пітливість і струшування ознобу
- Температура тіла нижче нормальної (у дорослих старше 65 років і у людей зі слабкою імунною системою)
- Нудота, блювота або діарея
- Задишка

У новонароджених і немовлят може не бути жодних ознак інфекції. Вони можуть блювати, мати лихоманку і кашель, може з'являтися неспокій або втома, пропажа енергії, труднощі з диханням і їжею.

Якщо у вас виникли труднощі з диханням, біль у грудях, стійка лихоманка 102 F (39 C) або вище, або постійний кашель, особливо якщо ви кашляєте гноєм, зверніться до лікаря.

Особливо важливо, щоб люди з цих груп підвищеного ризику звернулися до лікаря:

- Дорослі старші 65 років
- Діти молодші 2 років з ознаками і симптомами
- Люди з поганим станом здоров'я або ослабленою імунною системою
- Люди, які проходять курс хіміотерапії або приймають ліки, що пригнічують імунну систему

Для деяких літніх людей та людей з серцевою недостатністю або хронічними проблемами легень пневмонія може швидко стати загрозливим для життя станом.

Багато мікроб можуть викликати пневмонію. Найбільш поширеними є бактерії і віруси в повітрі, яким ми дихаємо. Людське тіло зазвичай запобігає зараженню цих мікробів за допомогою легень. Але іноді ці мікроби можуть перебороти вашу імунну систему, навіть якщо ваше здоров'я, зазвичай, добре.

Пневмонія класифікується відповідно до типів мікробів, що викликають її, і від того, де ви отримали інфекцію.

Найбільш поширеним видом пневмонії є позалікарняна пневмонія. Зараження відбувається поза лікарнями та іншими закладами охорони здоров'я. Це може бути викликано:

- Бактеріями. Найбільш частою причиною бактеріальної пневмонії в США є *Streptococcus pneumoniae*. Цей тип пневмонії може виникнути самостійно або після того, як ви простудилися або виникли грип. Вона може впливати на одну частину (частку) легені, стан, що називається лобарною пневмонією.

- Бактеріоподібні організми. *Mycoplasma pneumoniae* також може викликати пневмонію. Це, як правило, дає більш м'які симптоми, ніж інші види пневмонії. Ходяча пневмонія є неформальним ім'ям цього типу пневмонії, яке, як правило, не є достатньо серйозним, щоб вимагати постільного режиму.

- Гриби. Цей тип пневмонії найбільш часто зустрічається у людей з хронічними проблемами здоров'я або ослабленою імунною системою, а також у людей, які вдихали великі дози організмів. Гриби, що викликають його, можуть бути знайдені в ґрунті або пташиному посліді і змінюватися залежно від географічного розташування.

- Віруси. Деякі з вірусів, які викликають застуду і грип, можуть викликати пневмонію. Віруси є найпоширенішою причиною пневмонії у дітей молодше 5 років. Вірусна пневмонія зазвичай м'яка. Але в деяких випадках вона може стати дуже серйозною.

Деякі люди заражаються пневмонією під час перебування в стаціонарі на лікуванні іншої хвороби. Люди, що лежать в відділеннях інтенсивної терапії, піддаються підвищеному ризику цього типу пневмонії.

Такий тип пневмонії може бути більш небезпечним, адже бактерії, які спричинюють хворобу можуть бути вже більше стійкими до антибіотиків, а також тому що, люди, які лежать в стаціонарі вже є хворими і їхній організм є більш чутливим до збудників інших хворіб.

Також пневмонією можуть захворіти люди, що живуть у закладах довгострокового догляду або отримують допомогу в амбулаторних клініках, включаючи центри діалізу нирок. Подібно пневмонії, якою можна заразитись на стаціонарі, цей тип пневмонії також може бути викликаний бактеріями, які є більш стійкі до антибіотиків.

Аспіраційна пневмонія виникає при вдиханні частинок їжі, напоїв, блювоти або слини в легені. Аспірація є більш імовірною, якщо щось порушує ваш нормальний рефлекс, наприклад, травма мозку або проблема ковтання, або надмірне вживання алкоголю або наркотиків.

1.3.2 Фактори ризику

Пневмонією може захворіти кожен. Але виділяють дві вікові групи з найвищим ризиком [5]:

- Діти віком до 2 років
- Люди віком від 65 років

Інші фактори ризику включають:

- Госпіталізовані люди. Ви перебуваєте в групі підвищеного ризику пневмонії, якщо ви перебуваєте у відділенні інтенсивної терапії в лікарні, особливо якщо ви перебуваєте на машині, яка допомагає дихати (вентилятор).
- Хронічне захворювання. Ви частіше хворієте пневмонією, якщо у вас астма, хронічна обструктивна хвороба легень (ХОЗЛ) або хвороба серця.
- Куріння. Куріння завдає шкоди природному захисту організму від бактерій і вірусів, що викликають пневмонію.
- Ослаблена або пригнічена імунна система. Люди, які мають ВІЛ / СНІД, які перенесли трансплантацію органів, які проходять курс хіміотерапії або приймають довгострокові стероїди, знаходяться під загрозою.

1.3.3 Ускладнення

Навіть при лікуванні деякі люди з пневмонією, особливо у групах підвищеного ризику, можуть відчувати ускладнення, включаючи:

- Бактерії в крові (бактеріємія). Бактерії, які потрапляють у кров з легень, можуть поширювати інфекцію в інші органи.
- Труднощі дихання. Якщо пневмонія сильна або у пацієнта є хронічні захворювання легень, у нього можуть виникнути проблеми з вдиханням достатньої кількості кисню. Можливо, знадобиться

госпіталізація і використання дихального апарату (вентилятор) під час загоєння легень.

- Накопичення рідини навколо легень (плевральний випіт). Пневмонія може спричинити виникнення рідини в тонкому просторі між шарами тканини, що вистилають легені і грудної порожнини (плеври). Якщо рідина інфікується, вам може знадобитися її зливання через грудну трубку або видалення з операцією.

- Абсцес легень. Абсцес виникає, якщо гній утворюється в порожнині в легені. Абсцес зазвичай лікують антибіотиками. Іноді для видалення гною необхідно хірургічне втручання або дренивання довгою голкою чи трубкою, поміщеної в абсцес.

1.3.4 Профілактика

Щоб запобігти пневмонії:

- Щеплення. Вакцини доступні для запобігання деяких видів пневмонії та грипу. Рекомендації щодо вакцинації змінювалися з часом, тому необхідно з часом перевіряти статус вакцинації пацієнта, навіть якщо раніше він вже приймав ліки.

- Необхідно вакцинувати дітей. Лікарі рекомендують іншу вакцину від пневмонії для дітей молодше 2-х років і дітей віком від 2 до 5 років, які мають особливий ризик пневмококової інфекції. Діти, які відвідують групу дитячого догляду, також повинні отримати вакцину. Лікарі також рекомендують постріли грипу для дітей старше 6 місяців.

- Гігієна. Щоб захистити себе від респіраторних інфекцій, які іноді призводять до пневмонії, потрібно регулярно мити руки або використовувати дезінфікуючий засіб для рук на спиртовій основі.

- Відмова від куріння. Куріння завдає шкоди природним засобам захисту легенів від респіраторних інфекцій.

- Зберігайте імунну систему сильною. Висипайтесь, регулярно займайтеся фізичними вправами і харчуйтеся здоровою їжею.

1.4 Застосування технології машинного навчання для діагностування пневмонії

Глибинне навчання є галуззю штучного інтелекту, де мережі з простих з'єднаних між собою підрозділів використовуються для вилучення шаблонів з даних для вирішення складних проблем. Глибинні навчальні алгоритми показали новаторську продуктивність у різноманітних складних завданнях, особливо тих, які стосуються зображень. Вони часто співпадають або перевищують продуктивність людини. Оскільки тематична область радіології здебільшого спирається на вилучення корисної інформації з зображень, вона є природньою прикладною областю для глибинного навчання, і дослідження в цій області швидко зростали в останні роки.

Область глибинного навчання охоплює групу методів штучного інтелекту, які використовують велику кількість простих взаємопов'язаних одиниць для виконання складних завдань. Алгоритми глибинного навчання, замість використання набору попередньо запрограмованих інструкцій, здатні вчитися з великих обсягів даних. Завдання, які вирішуються цими алгоритмами, включають локалізацію та класифікацію об'єктів у зображеннях, розуміння мови, ігор та багато інших. Хоча флагман глибокого навчання, згорткові нейронні мережі, вперше були введені десятиліттями тому, лише останні 5 років продемонстрували дивовижний успіх цих алгоритмів, піднявши свій статус з цікавої, але зовсім непрактичної ідеї до алгоритму штучного інтелекту. В останні роки не тільки алгоритми глибинного навчання змогли перевершити продуктивність інших методів штучного інтелекту, але в деяких завданнях вони показали продуктивність, що перевершує людей.

Рентгенологія є галуззю медицини, яка зосереджена на використанні медичних зображень для виявлення, діагностики та характеристики захворювань (діагностичної радіології), а також для проведення процедурних втручань (інтервенційної радіології). У Сполучених Штатах типовий рентгенолог проходить навчання 14-15 років після закінчення середньої школи, включаючи 4 роки коледжу, 4 роки медичної школи, 1 рік стажування, 4 роки.

Першочерговим обов'язком рентгенолога є перегляд зображення, доставленого до його чи її станції читання, і формування письмового звіту. Ця добре структурована і ізольована природа роботи радіолога робить особливо привабливим застосування алгоритмів штучного інтелекту.

1.4.1 Глибинне навчання проти «традиційного» машинного навчання

Все частіше ми чуємо про відмінність між глибоким навчанням і «традиційним» машинним навчанням (див. Рис. 1.1). Різниця дуже важлива, особливо в контексті медичної візуалізації. В традиційному машинному навчанні, типовим першим кроком є вилучення ознак. Це означає, що для класифікації об'єкта необхідно вирішити, які характеристики об'єкта будуть важливі і реалізувати алгоритми, які здатні захопити ці характеристики [6]. Для цієї мети було запропоновано ряд складних алгоритмів у сфері комп'ютерного зору і вилучено різноманітність розмірів, форми, текстури та інших особливостей. Цей процес значною мірою є довільним, оскільки дослідник або практик, що вивчає машину, часто повинні здогадуватися, які особливості будуть корисні для конкретного завдання, і ризикує включати в себе непотрібні та надлишкові функції і, що ще важливіше, не дуже корисні функції.

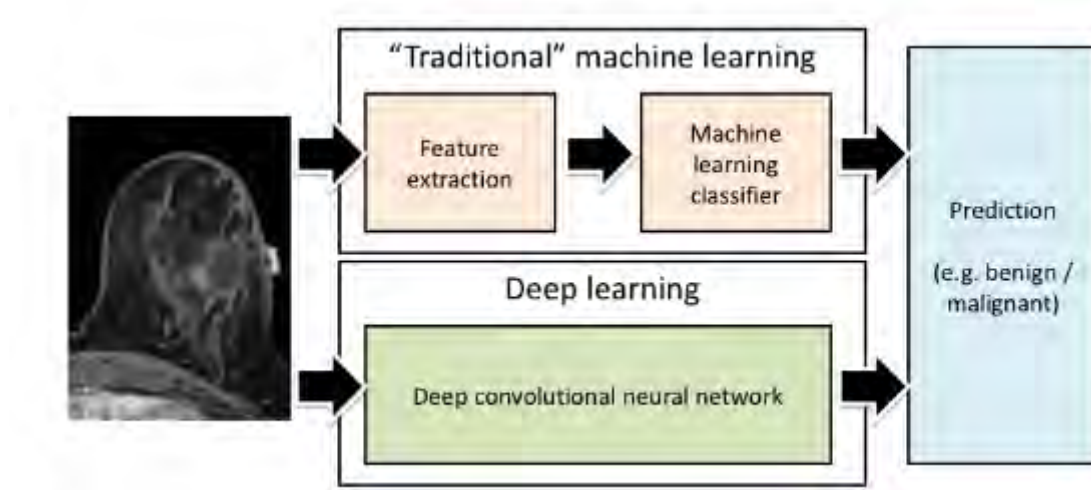


Рис. 1.1 «Традиційне» та глибинне машинне навчання

У процесі глибинного навчання процес вилучення ознак і прийняття рішень є зарубіжними і навчальними, і тому не потрібно робити жодного вибору щодо того, які функції слід витягувати; це вирішується мережею в

процесі навчання. Однак, вартість надання нейронної мережі для вибору власних функцій є вимогою для набагато більших наборів даних навчання.

1.4.2 Глибинне машинне навчання

Методи глибинного навчання (і алгоритми штучного інтелекту в цілому) мають величезний потенціал впливати на практику радіології. На відміну від більшості інших аспектів медицини, всі або майже всі необхідні елементи, що застосовують для поставлення діагнозу у радіології, можуть бути цифровими, що дозволяє аналізувати їх за допомогою штучних інтелектуальних алгоритмів.

Існує загальна домовленість, що глибинне навчання буде відігравати певну роль у майбутній практиці радіології. Деякі передбачають, що він буде проводити повсякденні завдання, залишаючи радіологам більше часу, щоб зосередитися на інтелектуально складних викликах. Деякі вважають, що рентгенологи та алгоритми глибокого навчання працюватимуть разом, щоб забезпечити кращу продуктивність, ніж будь-який з них окремо взятий. Нарешті, існує прогноз, що глибинні алгоритми навчання замінять рентгенологів взагалі (принаймні в їх можливості інтерпретації зображення).

Включення глибинного навчання в радіологію буде пов'язано з численними викликами. По-перше, і в першу чергу, це технологічний виклик. Хоча глибинне навчання показало надзвичайні успіхи в інших завданнях, пов'язаних з зображенням, результати в рентгенології ще зовсім не показують, що глибинні алгоритми навчання замінять рентгенолога у всьому обсязі їх діагностичної роботи. Деякі останні дослідження свідчать про виконання цих алгоритмів в порівнянні з експертом людини, але ці результати застосовні лише для дуже невеликої меншості завдань, які виконують рентгенологи. Це, швидше за все, зміниться в наступні роки з огляду на швидкий прогрес у реалізації глибинних алгоритмів навчання в області радіології

Впровадження глибинного навчання в радіологічній практиці також створює правові та етичні проблеми. Перш за все: хто буде нести відповідальність за помилки, які комп'ютер зробить? Хоча це важке питання,

подібні питання ставилися і вирішувалися, коли були впроваджені інші технології, включаючи ліфти та автомобілі. Оскільки штучний інтелект проникає в різні сфери діяльності людини, питання такого типу, швидше за все, будуть вивчені і відповіді запропоновані в найближчі роки

Інші виклики включатимуть прийняття пацієнта або неприйняття радіолога, який не бере участь у процесі інтерпретації їхніх зображень (незалежно від виконання), а також регуляторних питань. Нарешті, важливим практичним питанням є те, як включити глибинні алгоритми навчання в радіологічний робочий процес, щоб поліпшити, а не погіршити радіологічну практику.

2. Огляд машинного навчання

2.1 Визначення нейронної мережі

Нейронні мережі є набором алгоритмів, змодельованих на основі мозку людини, призначених для розпізнавання закономірностей [7]. Вони інтерпретують сенсорні дані за допомогою свого роду машинного сприйняття, маркування або кластеризації вхідних даних. Закономірності, які розпізнають нейронні мережі, є числовими, містяться у векторах, в які повинні бути перекладені всі дані реального світу, чи то зображення, звук, текст або часові ряди.

Нейронні мережі допомагають нам кластеризувати і класифікувати. Можна вважати їх шаром кластеризації та класифікації поверх даних, які ви зберігаєте та якими оперуєте. Вони допомагають групувати нерозмічені дані відповідно до схожості між прикладами вхідних даних, і класифікують дані, коли вони мають нанесений набір даних для навчання. (Нейронні мережі можуть також витягувати особливості, які надходять до інших алгоритмів кластеризації та класифікації, так що ви можете думати про глибинні нейронні мережі як про компоненти більших машино-навчальних додатків, що включають алгоритми навчання, класифікації та регресії).

Щоб зрозуміти чи можна використовувати глибинне навчання для певної проблеми, потрібно задати питання:

- На які результати я очікую? Ці результати є мітками, які можна застосувати до даних: наприклад, `spam` або `not_spam` у фільтрі електронної пошти, `good_guy` або `bad_guy` у виявленні шахрайства, `angry_customer` або `happy_customer` у керуванні відносинами з клієнтами.
- Чи є у мене дані, щоб супроводжувати ці мітки? Тобто, чи можна знайти позначені дані, або чи я можу створити маркований набір даних (з такою послугою, як `AWS Mechanical Turk` чи `Figure Eight` чи `Mighty.ai`), де спам був позначений як спам, щоб навчити алгоритм кореляції між мітками і вхідними даними?

Глибоке навчання відображає входи на виходи. Воно знаходить кореляції і відоме як «універсальний апроксиматор», оскільки може навчитися наближати невідому функцію $f(x) = y$ між будь-яким входом x і будь-яким виходом y , припускаючи, що вони пов'язані (наприклад, кореляцією чи причиною). У процесі навчання нейронна мережа знаходить правильне f , або правильний спосіб перетворення x в y , будь то $f(x) = 3x + 12$ або $f(x) = 9x - 0.1$. Ось кілька прикладів того, що може зробити глибоке навчання.

2.2 Класифікація

Всі завдання класифікації залежать від позначених наборів даних; тобто люди повинні передати свої знання набору даних для того, щоб нейронна мережа дізналася кореляцію між мітками та даними. Такий спосіб машинного навчання відомий як навчання з учителем.

- Виявлення обличчя, ідентифікація людей у зображеннях, розпізнавання міміки (сердитий, радісний)
- Визначення об'єктів на зображеннях (знаки зупинки, пішоходи, маркери доріжок)
- Розпізнавання жестів у відео
- Виявлення голосів, ідентифікація людей, що говорять, розпізнавання мови, розпізнавати почуття голосами
- Класифікація тексту як спам (електронною поштою) або шахрайським (у страхових випадках); розпізнавання настрою в тексті (відгуки клієнтів)

Будь-які мітки, які люди можуть генерувати, будь-які результати, які вам необхідні і які при цьому корелюють з даними, можна використовувати для навчання нейронної мережі.

2.3 Кластеризація

Кластеризація або групування - це виявлення подібності. Глибоке навчання не вимагає маркування для виявлення подібності. Навчання без позначок називається без вчителя. Нерозмічені дані - це більшість даних у

світі. Одним із законів машинного навчання є: чим більше даних тренується алгоритмом, тим точніше він буде. Таким чином, навчання без вчителя має потенціал для створення високоточних моделей.

- Пошук: порівняння документів, зображень або звуків на поверхні подібних елементів.
- Виявлення аномалій: в іншому випадку виявлення подібностей виявляє аномалії або незвичайну поведінку. У багатьох випадках незвичайна поведінка високо корелює з речами, які ви хочете виявити і запобігти, наприклад, шахрайству.

2.4 Регресійний аналіз

За допомогою класифікації, глибинне навчання здатне встановити кореляції між, скажімо, пікселями в зображенні і імям людини. Ви можете назвати це статичним прогнозом. До того ж, піддавшись достатній кількості правильних даних, глибинне навчання здатне встановити кореляцію між поточними подіями та майбутніми подіями. Це може призвести до регресії між минулим і майбутнім. Глибинне навчання не обов'язково піклується про час, або той факт, що щось ще не відбулося. Враховуючи часові ряди, глибинне навчання може читати рядок чисел і передбачати число, яке найімовірніше відбудеться далі.

- Поломки обладнання (центри обробки даних, виробництво, транспорт)
- Хвороби (інсульт, серцеві напади на основі життєво важливої статистики та дані з носія)
- Відтік клієнтів (прогнозування ймовірності виходу клієнта на основі веб-активності та метаданих)
- Обороти працівника (те ж саме, але для співробітників)

Чим краще ми можемо передбачити проблему, тим краще ми можемо її запобігти. Як ви можете бачити, з нейронними мережами ми рухаємося з меншою кількістю сюрпризів. Ми також рухаємося до світу розумніших

агентів, які поєднують нейронні мережі з іншими алгоритмами, такими як навчання з підкріпленням для досягнення цілей.

Після короткого огляду варіантів використання глибинного навчання, давайте подивимося, з чого складаються нейронні мережі.

2.5 Елементи нейронної мережі

Глибинне навчання - це ім'я, яке ми використовуємо для «стекових нейронних мереж»; тобто мережі, що складаються з декількох шарів.

Шари виконані з вузлів. Вузол це просто місце, де відбувається обчислення (рис. 2.1). Він утворений на основі нейрона людського мозку, який запускається, коли він стикається з достатніми стимулами. Вузол поєднує вхідні дані з набором коефіцієнтів або ваг, які або посилюють, або зменшують цей вхід, тим самим призначаючи значення входам щодо завдання, яке алгоритм намагається дізнатися. Наприклад, який вхід найбільш корисний для класифікації даних без помилок? Ці продукти вхідної маси підсумовуються, а потім сума передається через так звану функцію активації вузла, щоб визначити, як цей сигнал повинен просуватися далі через мережу, щоб вплинути на кінцевий результат, наприклад, акт класифікації. Якщо сигнали проходять, нейрон «активується» [1, 7].

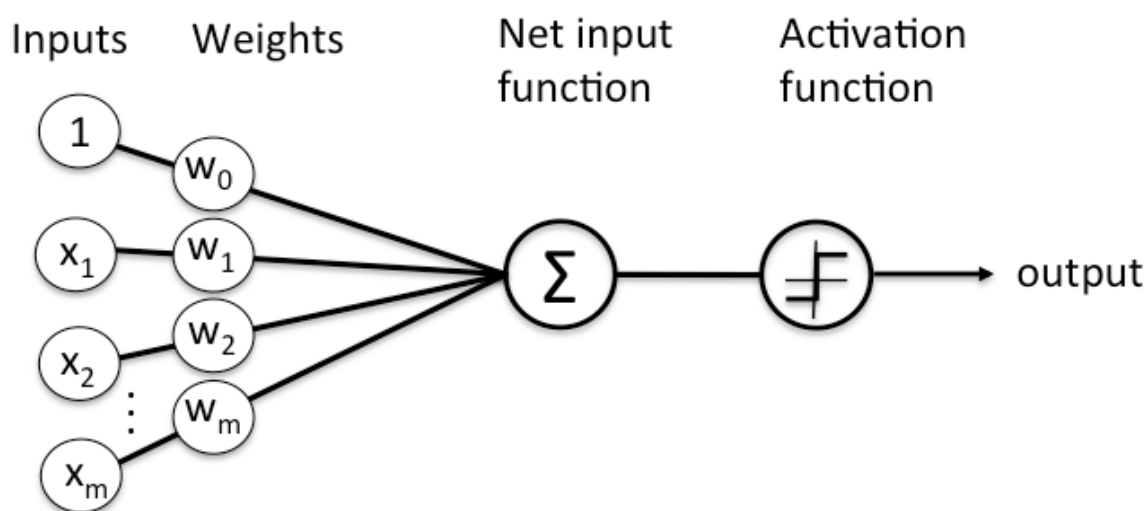


Рис. 2.1. Один вузол нейронної мережі

Шар вузла - це ряд тих нейроно-подібних комутаторів, які включаються або вимикаються, коли вхід подається через мережу. Вихід кожного шару одночасно є входом наступного шару, починаючи з початкового рівня введення, який отримує ваші дані (рис. 2.2).

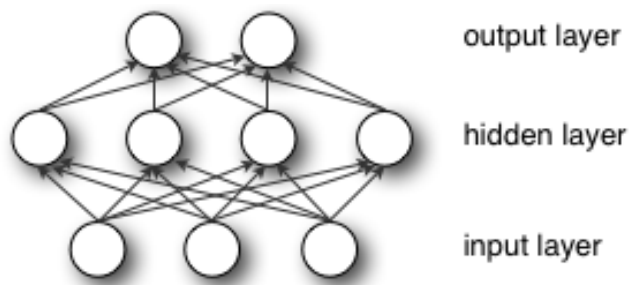


Рис. 2.2 Структурна схема шарів нейронної мережі

Поєднання регульованих ваг моделі з вхідними функціями це те, як ми приписуємо значення цим особливостям щодо того, як нейронна мережа класифікує та вводить кластери.

2.6 Ключові концепції нейронних мереж

Глибинні навчальні мережі відрізняються від більш загальноприйнятих нейронних мереж з одним прихованим шаром за їх глибиною; число вузлових шарів, через які дані повинні проходити в багатоетапному процесі розпізнавання образів.

Більш ранні версії нейронних мереж, такі як перші перцептрони, були неглибокі, склалися з одного вхідного і одного вихідного шару, і не більше одного прихованого шару між ними. Більше трьох шарів (включаючи вхід і вихід) кваліфікується як «глибинне» навчання. Отже, глибинне це не просто модне слово, а строго визначений термін, що означає більше ніж один прихований шар.

У мережах глибокого навчання кожен шар вузлів тренується на певному наборі функцій на основі вихідних даних попереднього шару (рис. 2.3). Чим далі ви просуваєтеся в нейронну мережу, тим складніші функції ваших вузлів

можуть розпізнавати, оскільки вони об'єднують і рекомбінують функції з попереднього шару.

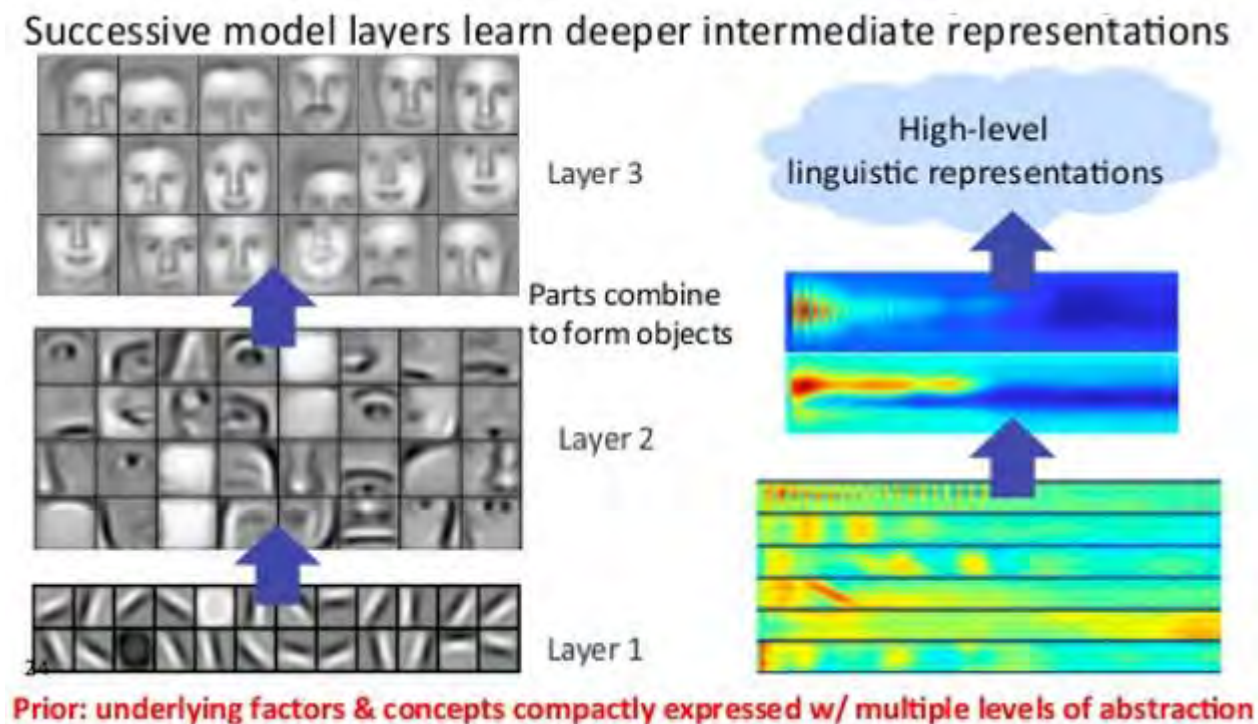


Рис. 2.3 Ілюстративне зображення ієрархії функцій нейронної мережі

Це називається ієрархією функцій, тобто ієрархією зростаючої складності та абстракції. Завдяки цьому мережі глибинного навчання здатні обробляти дуже великі, високовимірні набори даних з мільярдами параметрів, які проходять через нелінійні функції.

Перш за все, ці нейронні мережі здатні виявляти латентні структури в межах нерозмічених, неструктурованих даних, які, по суті, переважають у світі. Іншими словами для неструктурованих даних є багато джерел: фотографії, тексти, відео та аудіозаписи. Таким чином, одна з проблем, яку глибоке навчання вирішує найкраще, полягає в обробці та кластеризації необроблених, немаркованих у світі засобів масової інформації, що розрізняють подібності та аномалії в даних, які жодна людина не організувала в реляційній базі даних, або ніколи не проіменувала.

Наприклад, глибинне навчання може займати мільйон зображень, і кластеризувати їх відповідно до їх схожості: кішки в одному кутку, криголами в іншому, і в третій всі фотографії вашої бабусі. Це є основою так званих смартфонів-фотоальбомів.

Тепер застосуємо ту ж саму ідею до інших типів даних: глибинне навчання може кластерувати необроблений текст, наприклад, електронні листи або новини. Електронні листи, повні роздратованих скарг, можуть кластеруватися в одному кутку векторного простору, тоді як задоволені клієнти або спам-боти можуть класти в інших. Це є основою різних фільтрів обміну повідомленнями і може бути використано в управлінні відносинами з клієнтами (CRM). Те ж саме стосується і голосових повідомлень.

З часовими рядами дані можуть групуватися навколо нормальної/здорової поведінки та аномальної/небезпечної поведінки. Якщо дані часових рядів генеруються смартфоном, це дасть змогу зрозуміти здоров'я користувачів і звички; якщо воно генерується автозапуском, воно може бути використано для запобігання катастрофічних збоїв.

Глибинні навчальні мережі здійснюють автоматичне вилучення функцій без втручання людини, на відміну від більшості традиційних алгоритмів машинного навчання. Беручи до уваги, що вилучення ознак є завданням, яке може зайняти роки, якщо обчислення буде проводити група вчених, глибинне навчання - це спосіб обійти обмежену кількість експертів. Це збільшує повноваження невеликих науково-дослідних команд, які за своєю природою не масштабуються.

При навчанні на нерозмічених даних кожен шар вузла в глибинній мережі вивчає функції автоматично, неодноразово намагаючись відновити вхід, з якого він малює свої зразки, намагаючись звести до мінімуму різницю між здогадами мережі і розподілом ймовірностей вхідних даних. Обмежені машини Больцмана, наприклад, створюють так звані реконструкції таким чином.

У процесі роботи ці нейронні мережі вчаться розпізнавати кореляції між деякими відповідними ознаками та оптимальними результатами - вони з'єднують сигнал характеристики і те, що представляє собою ця характеристика, поки це не буде повною реконструкцією або позначеними даними.

Глибинна навчальна мережа, підготовлена з маркованих даних, може бути застосована до неструктурованих даних, надаючи їй доступ до набагато більшого вхідного матеріалу, ніж мережеві засоби машинного навчання. Це рецепт більш високої продуктивності: чим більше даних тренується мережею, тим точніше вона буде. (Погані алгоритми, що вивчаються на великій кількості даних, можуть перевершити хороші алгоритми, які тренувалися дуже мало.) Здатність глибинного навчання обробляти та вивчати величезну кількість нерозмічених даних дає їм чітку перевагу над попередніми алгоритмами.

Глибинні навчальні мережі закінчуються вихідним шаром: logistic або softmax класифікатор, який призначає ймовірність певного результату або мітки. Ми називаємо це передбачуванним, але він передбачає в широкому сенсі. Враховуючи необроблені дані у вигляді зображення, мережа глибокого навчання може вирішити, наприклад, що вхідні дані на 90 відсотків представляють людину.

2.7 Мережі "Feedforward"

Наша мета у використанні нейронної мережі полягає в тому, щоб прийти до точки найменшої помилки якнайшвидше. Ми ведемо перегони, і гонка навколо доріжки, тому ми проходимо ті ж самі пункти неодноразово в циклі. Стартовою лінією гонки є стан, в якому ініціалізуються наші ваги, а фінішний рядок - стан цих параметрів, коли вони здатні виробляти достатньо точні класифікації та прогнози.

Сама дистанція включає багато кроків, і кожен з цих кроків нагадує кроки до і після. Так само, як бігун, ми будемо займатися повторюваним актом знову і знову, щоб дістатися до фінішу. Кожен крок для нейронної мережі передбачає

припущення, вимірювання помилок і незначне оновлення його ваг, інкрементне коригування коефіцієнтів, оскільки він повільно вивчає найважливіші особливості.

Колекція ваг, незалежно від того, чи вони перебувають у стартовому або кінцевому стані, також називається моделлю, оскільки вона є спробою моделювання відношення даних до міток, щоб зрозуміти структуру даних. Моделі зазвичай починають погано і в кінцевому підсумку стають кращими, змінюючись з часом і оновлюючи параметри нейронної мережі.

Це пояснюється тим, що нейронна мережа народжується в невігластві. Вона не знає, які ваги і зміщення переведуть вхідні дані в правильні висновки. Потрібно починати з припущення, а потім намагатися послідовно покращувати результат, навчаючись на помилках. (Можна думати про нейронну мережу як про мініатюрну постановку наукового методу випробування гіпотез і спроби знову - лише науковий метод із зав'язаними очима. Або як дитина: вона народжується не знаючи багато, і через вплив життєвого досвіду, вона повільно навчається вирішувати проблеми у світі).

Ось просте пояснення того, що відбувається під час навчання з нейронною мережею.

Вхідні дані надходять у мережу. Коефіцієнти або вагові коефіцієнти, які вводяться до множини припущень, що робить мережа в кінці.

$$input * weight = guess \quad (2.1)$$

Результати зважених вхідних даних угадують, що таке вхідні дані. Після цього нейрон отримує свою думку і порівнює її з вказаною правдивою інформацією про дані, фактично запитуючи експерта "Чи я правий?"

$$ground\ truth - guess = error \quad (2.2)$$

Різниця між припущенням мережі та основною істиною - це її помилка. Мережа вимірює цю помилку і повертає помилку назад по своїй моделі, коригуючи ваги в тій мірі, в якій вони сприяли помилці.

$$error * weight's contribution to error = adjustment \quad (2.3)$$

Три псевдо-математичні формули вище враховують три ключові функції нейронних мереж: оцінювання вхідних даних, обчислення втрат і застосування оновлення до моделі - щоб почати триступеневий процес знову. Нейронна мережа - це коректуючий контур зворотного зв'язку, нагороджуючи вагами, що підтримують його правильні здогади, і караючи вагами, які призводять до помилки.

2.8 Множинна лінійна регресія

Незважаючи на свою біологічно натхненну назву, штучні нейронні мережі є не чим іншим, як математикою і кодом, як і будь-який інший алгоритм машинного навчання. Насправді, кожен, хто розуміє лінійну регресію, один з перших методів, який ви вивчаєте в статистиці, може зрозуміти, як працює нейронна мережа. У найпростішій формі лінійна регресія виражається як

$$Y_{hat} = bX + a \quad (2.4)$$

де Y_{hat} є розрахунковим результатом, X є входом, b - нахилом і a - переходом лінії на вертикальній осі двовимірного графа. (Щоб зробити це більш конкретним: X може бути радіаційним опроміненням і Y може бути ризиком раку; X може бути щоденним віджиманням і Y_{hat} може бути загальною вагою, яку можна встановити; X - кількість добрив і Y_{hat} розмір культури). Можна собі уявити, що кожен раз, коли ви додаєте одиницю до X , залежна змінна Y_{hat} зростає пропорційно, незалежно від того, наскільки далеко ви знаходитесь на осі X . Це просте співвідношення між двома змінними, що рухаються вгору або вниз разом, є відправною точкою.

Наступним кроком є уявлення про множинну лінійну регресію, де у вас є багато вхідних змінних, які створюють вихідну змінну. Зазвичай це виражається так:

$$Y_{hat} = b_1 * X_1 + b_2 * X_2 + b_3 * X_3 + a \quad (2.5)$$

(Щоб розширити приклад врожаю вище, можна додати кількість сонячного світла і кількості опадів у вегетаційний період до змінної добрива, причому всі три змінні впливають на Y_{hat} .)

Тепер ця форма множинної лінійної регресії відбувається на кожному вузлі нейронної мережі. Для кожного вузла одного шару введення з кожного вузла попереднього шару рекомбінується з входом від кожного іншого вузла. Тобто входи змішуються в різних пропорціях, відповідно до їх коефіцієнтів, які є різними для кожного вузла наступного шару. Таким чином, мережа тестує, яка комбінація вхідних даних є значною, оскільки вона намагається зменшити помилку.

Після того, як ви підсумуєте вхідні дані вузла, щоб прийти до Y_{hat} , він передається через нелінійну функцію. Ось чому: якщо кожен вузол просто виконує множинну лінійну регресію, Y_{hat} збільшуватиметься лінійно і без обмежень, оскільки збільшиться X , але це не відповідає нашим цілям.

Те, що ми намагаємося побудувати на кожному вузлі, це перемикач, який вмикається і вимикається, залежно від того, чи повинен він дозволити сигналу входу пройти, щоб вплинути на кінцеві рішення мережі.

Якщо у нас є перемикач, у нас є проблема класифікації. Чи вказує вхідний сигнал, що вузол повинен класифікувати його як `enough`, або `not_enough`, включений або вимкнений? Двійкове рішення може бути виражене 1 і 0, а логістична регресія - нелінійна функція, яка скасовує вхідний сигнал, щоб перевести його в простір між 0 і 1.

Нелінійні перетворення в кожному вузлі зазвичай є S-подібними функціями, подібними до логістичної регресії. Вони йдуть по іменах `sigmoid` (грецьке слово для "S"), `tanh`, `hard tanh` і т.д., і вони формують вихід кожного вузла. Вихідні дані всіх вузлів, кожен з яких стискається в S-подібний простір між 0 і 1, потім передаються як вхідні дані для наступного шару в нейромережу

вперед, і так далі, поки сигнал не досягне кінцевого шару мережі, де приймаються рішення.

2.9 Градієнтний спуск

Градiєнтний спуск це одна з часто використовуваних функцій оптимізації, яка регулює ваги відповідно до викликаної ними помилки.

Градiєнт - це інше слово для нахилу, а нахил у типовій формі на графіку ху це те, як дві змінні відносяться один до одного: підвищення над пробігом, зміна грошей над зміною у часі, тощо. Нахил, який нас цікавить, описує взаємозв'язок між помилкою мережі і однією вагою, тобто те, як змінюється похибка при зміні ваги.

Поки нейронна мережа навчається, вона повільно налаштовує багато ваг, щоб вони могли правильно відобразити сигнал. Взаємозв'язок між помилкою мережі та кожною з цих ваг є похідною $\frac{dE}{dw}$, яка вимірює ступінь, до якої невелика зміна ваги викликає невелику зміну помилки.

Кожна вага - це лише один з факторів у глибинній мережі, що включає багато перетворень; сигнал ваги проходить через активації і підсумовує декілька шарів, тому ми використовуємо ланцюгове правило обчислення для того, щоб пройти через активації та виходи мереж і, нарешті, досягти поставленої ваги, а також її відношення до загальної помилки.

Правило ланцюга в обчисленні говорить, що

$$\frac{dz}{dx} = \frac{dz}{dy} * \frac{dy}{dx} \quad (2.6)$$

У мережі прямого доступу взаємозв'язок між помилкою мережі та однією вагою виглядає приблизно так:

$$\frac{dError}{dweight} = \frac{dError}{dactivation} * \frac{dactivation}{dweight} \quad (2.7)$$

Тобто, враховуючи дві змінні Error і weight, опосередковані третьою змінною, activation, через яку проходить weight, можна обчислити, як зміна weight впливає на зміну Error, спочатку обчислюючи, як впливає зміна activation впливає на зміну Error та як зміна weight впливає на зміну activation.

Суть навчання в глибокому навчанні - це не що інше як коригування ваги моделі у відповідь на помилку, яку вона виробляє, поки ви більше не зможете зменшити помилку.

Деякі приклади алгоритмів оптимізації:

- ADADELTA
- ADAGRAD
- ADAM
- NESTEROVS
- NONE
- RMSPROP
- SGD
- CONJUGATE GRADIENT
- HESSIAN FREE
- LBFGS
- LINE GRADIENT DESCENT

2.10 Функції активації

Функція активації визначає вихід, який буде генерувати вузол, на основі його входу.

Деякі приклади функцій активації:

- CUBE
- ELU
- HARDSIGMOID
- HARDTANH

- IDENTITY
- LEAKYRELU
- RATIONALTANH
- RELU
- RRELU
- SIGMOID
- SOFTMAX
- SOFTPLUS
- SOFTSIGN
- TANH

2.11 Логістична регресія

На глибинній нейронній мережі багатьох шарів кінцевий шар має особливу роль. При роботі з маркованим входом, вихідний шар класифікує кожен приклад, застосовуючи найбільш ймовірну мітку. Кожен вузол на вихідному шарі являє собою одну мітку, і цей вузол вмикається або вимикається відповідно до сили сигналу, який він отримує від входу і параметрів попереднього шару.

Кожен вихідний вузол виробляє два можливих результату, значення бінарного виходу 0 або 1, оскільки вхідна змінна або заслуговує мітки, або ні.

Хоча нейронні мережі, що працюють з розміченими даними, створюють двійковий вихід, вхід, який вони отримують, часто є безперервним. Тобто сигнали, які мережа отримує як вхідні дані, охоплюватимуть діапазон значень і включатимуть будь-яку кількість показників, залежно від проблеми, яку вона прагне вирішити.

Механізм, який ми використовуємо для перетворення безперервних сигналів у бінарний вихід, називається логістичною регресією. Назва не приємна, оскільки логістична регресія використовується для класифікації, а не

регресії в лінійному сенсі, з яким більшість людей знайомі. Вона обчислює ймовірність того, що набір входів відповідає позначці.

$$F(x) = \frac{1}{1+e^{-x}} \quad (2.8)$$

Для того, щоб безперервні входи були виражені у вигляді ймовірностей, вони повинні давати позитивні результати, оскільки не існує такої речі, як негативна ймовірність. Ось чому ви бачите вхід як показник e в знаменнику - тому що експоненти змушують наші результати бути більше нуля. Тепер розглянемо відношення експоненти e до фракції $1/1$. Один, як ми знаємо, є стелею ймовірності, за якою наші результати не можуть бути безглуздими.

Оскільки вхід x , який запускає мітку, зростає, вираз e до x зменшується до нуля, залишаючи нам частку $1/1$, або 100%, що означає, що ми наближаємося (не досягаючи) абсолютної впевненості, що етикетка застосовується. Вхідні дані, які негативно корелюють з вашим виходом, матимуть значення, відхилене негативним знаком експоненти e , і коли цей негативний сигнал зростає, величина e до x зростає, просуваючи всю фракцію все ближче до нуля.

Тепер уявімо, що замість того, щоб мати x як показник, у вас є сума продуктів всіх ваг та їх відповідних входів - сумарний сигнал, що проходить через вашу мережу. Це те, що ви подаєте до шару логістичної регресії на вихідному шарі класифікатора нейронної мережі.

За допомогою цього шару ми можемо встановити поріг рішення, над яким приклад позначений 1, а нижче якого він не є. Ви можете встановити різні пороги, як ви віддаєте перевагу - низький поріг збільшить кількість помилкових спрацьовувань, а більш високий - збільшить кількість помилкових негативів - залежно від того, на якій стороні ви хочете помилитися.

2.12 Нейронні мережі та штучний інтелект

У деяких колах нейронні мережі розглядаються як «груба сила» AI, оскільки вони починаються з чистого аркуша і забивають свій шлях до точної

моделі. Вони є ефективними, але на деякий погляд неефективними в їх підході до моделювання, які не можуть робити припущень щодо функціональних залежностей між виходом і входом.

Тим не менш, градієнтний спуск не рекомбінував кожен вагу з кожним іншим, щоб знайти кращий збіг - його метод пошуку шляхів скорочує відповідний ваговий простір, а отже, кількість оновлень і необхідних обчислень, на багато порядків. Більше того, такі алгоритми, як капсульні мережі Хінтона, вимагають набагато менше випадків даних для зближення на точній моделі; теперішнє дослідження має потенціал для вирішення грубої сили глибинного навчання.

2.13 Аналіз різних мов програмування для реалізації неронних мереж

2.13.1 Python

Python - це загальноприйнята мова програмування високого рівня, яка все частіше використовується в науці даних і в розробці алгоритмів машинного навчання. В даній мові програмування є багато бібліотек для роботи з великою кількістю даних, такі як: `numpy`, `scipy`, `pandas`, `matplotlib`. Він може бути застосований для розробки алгоритмів машинного навчання, які вирішують реальні світові проблеми. Дана мова програмування також охоплює всі важливі поняття, такі як дослідницький аналіз даних, попередня обробка даних, вилучення об'єктів, візуалізація даних і кластеризація, класифікація, регресія та оцінка продуктивності моделі. Також за допомогою Python можна створювати різні проекти з функціональними можливостями, таких як класифікація новин, виявлення спаму, онлайн-прогнозування кліків, прогноз цін на акції та інші важливі алгоритми машинного навчання.

Вважається, що екосистема Python добре підходить для проектів на основі AI. Python, з його простотою, великим співтовариством і інструментами, дозволяє розробникам створювати архітектури, близькі до досконалості, зберігаючи при цьому зосередженість на бізнес-керованих завданнях.

2.13.2 R

R зазвичай застосовується, коли потрібно аналізувати та маніпулювати даними для статистичних цілей. R має такі пакети, як Gmodels, Class, Tm і RODBC, які зазвичай використовуються для створення проектів машинного навчання. Ці пакети дозволяють розробникам реалізовувати алгоритми машинного навчання без зайвих клопотів і дозволяють швидко реалізувати бізнес-логіку.

R було створено статистиками для задоволення їхніх потреб. Ця мова може дати вам глибокий статистичний аналіз, як і для обробки даних з пристрою IoT так і для аналізу фінансової моделі.

Більше того, якщо ваше завдання вимагає високоякісних графіків і діаграм, ви можете використовувати R. За допомогою ggplot2, ggvis, googleVis, Shiny, rCharts та інших пакунків можливості R значно розширюються, допомагаючи перетворювати візуальні ефекти в інтерактивні веб-програми.

У порівнянні з Python, R має репутацію повільного і відсталого, коли мова йде про великомасштабні продукти даних. Краще використовувати Python або Java, з її гнучкістю, для реальної розробки продукту.

2.13.3 Scala

Scala є безцінною, коли справа доходить до великих даних. Вона пропонує вченим масив таких інструментів, як сідло, скалалаб і бриз. Scala має велику підтримку паралелізму, яка допомагає обробляти великі обсяги даних. Оскільки Scala працює на JVM, вона виходить за рамки всіх обмежень разом з Nadoop, відкритим джерелом розподіленої обробки, що управляє обробкою та зберіганням даних для великих додатків даних, що працюють у кластерних системах. Незважаючи на меншу кількість інструментів для машинного навчання, у порівнянні з Python і R, Scala є надзвичайно доступним.

2.13.4 Julia

Якщо вам потрібно побудувати рішення для високопродуктивних обчислень і аналізу, ви можете розглянути Julia. Julia має аналогічний

синтаксис для Python і була розроблена для обробки чисельних обчислювальних завдань. Julia надає підтримку для глибокого навчання через обгортку TensorFlow.jl і рамку Mocha.

Проте мова не підтримується багатьма бібліотеками і ще не має сильної спільноти на зразок Python, оскільки вона є відносно новою.

2.13.5 Java

Ще однією мовою, про яку варто згадати, є Java. Java є об'єктно-орієнтованою, портативною, підтримуваною та прозорою. Її підтримують численні бібліотеки, такі як WEKA і Rapidminer.

Java широко поширена, коли мова йде про обробку природних мов, алгоритми пошуку та нейронні мережі. Це дозволяє швидко будувати великомасштабні системи з відмінною продуктивністю.

Але якщо ви хочете виконати статистичне моделювання та візуалізацію, то Java є останньою мовою, яку ви хочете використовувати. Незважаючи на наявність деяких пакетів Java, які підтримують статистичне моделювання та візуалізацію, вони не є достатніми. Тоді як Python має розширені інструменти, які добре підтримуються спільнотою.

2.14 Python як найкраща мова для розробки штучного інтелекту

Проекти штучного інтелекту відрізняються від традиційних програмних проєктів. Відмінності полягають у технології стеку, навичках, необхідних для проєкту на основі ШІ, і необхідності глибоких досліджень. Щоб реалізувати свої прагнення до ШІ, ви повинні використовувати мову програмування, яка є стабільною, гнучкою і має доступні інструменти.

Від розробки до розгортання та обслуговування, Python допомагає розробникам бути продуктивними та впевненими у створенні програмного забезпечення. Переваги, які роблять Python найбільш придатними для машинного навчання та проєктів на основі AI, включають простоту та послідовність, доступ до великих бібліотек та структур для штучного інтелекту

та машинного навчання, гнучкість, незалежність платформи та широке співтовариство. Вони додають загальної популярності мови.

Python пропонує стислий і читабельний код. Незважаючи на складні алгоритми та різноманітні робочі процеси, що стоять за машинним навчанням та ШІ, простота Python дозволяє розробникам писати надійні системи. Розробники докладають всіх зусиль для вирішення проблеми машинного навчання, замість того, щоб зосередитися на технічних нюансах мови.

Крім того, Python є привабливим для багатьох розробників, оскільки його легко вивчити. Код Python зрозумілий людям, що полегшує створення моделей для машинного навчання.

Багато програмістів кажуть, що Python є більш інтуїтивним, ніж інші мови програмування. Інші вказують на безліч рамок, бібліотек і розширень, які спрощують реалізацію різних функціональних можливостей. Загальноприйнято, що Python підходить для спільної реалізації, коли задіяно кілька розробників. Оскільки Python є мовою загального призначення, він може виконувати набір складних завдань машинного навчання і дозволяє швидко створювати прототипи, які дозволяють тестувати ваш продукт для цілей машинного навчання.

Впровадження алгоритмів ШІ і МН може бути складним і вимагає багато часу. Дуже важливо мати добре структуроване та добре перевірене оточення, щоб розробники могли придумати найкращі рішення для кодування.

Щоб скоротити час розробки, програмісти звертаються до кількох фреймворків і бібліотек Python. Бібліотека програмного забезпечення це попередньо написаний код, який розробники використовують для вирішення спільних завдань програмування. Python, з його багатим стеком технологій, має великий набір бібліотек для штучного інтелекту та машинного навчання. Ось деякі з них:

- Keras, TensorFlow, і Scikit-learn для машинного навчання

- NumPy для високопродуктивних обчислювань і аналізу даних
- SciPy для розширених обчислень
- Pandas для аналізу даних в загальних потребах
- Seaborn для візуалізації даних

Scikit-learn має різні алгоритми класифікації, регресії та кластеризації, включаючи векторні машини підтримки, випадкові ліси, градієнтні підсилення та DBSCAN, і призначений для роботи з чисельними та науковими бібліотеками Python та SciPy.

За допомогою цих рішень ви можете розвивати свій продукт швидше. Вашій команді розробників не доведеться винаходити велосипед, тому що можна використовувати існуючу бібліотеку для реалізації необхідних функцій.

Незалежність платформи відноситься до мови програмування або фреймворку, що дозволяє розробникам реалізовувати проекти на одній машині і використовувати їх на іншій машині без будь-яких (або з мінімальними) змін. Одним із ключових для популярності Python є те, що вона є незалежною від платформи мовою. Python підтримується багатьма платформами, включаючи Linux, Windows і macOS. Код Python може використовуватися для створення окремих виконуваних програм для більшості звичайних операційних систем, що означає, що програмне забезпечення Python може бути легко розповсюджено і використане на тих операційних системах без інтерпретатора Python.

Більш того, розробники зазвичай використовують такі послуги, як Google або Amazon, для своїх обчислювальних потреб. Тим не менш, ви часто можете знайти компанії і вчених, які використовують свої власні машини з потужними блоками обробки графіки (GPU) для навчання своїх моделей МН. І той факт, що Python є незалежним від платформи, робить цей тренінг набагато дешевшим і простішим.

У огляді розробників 2018 від Stack Overflow, Python входив до десятки найпопулярніших мов програмування, що в кінцевому підсумку означає, що ви можете знайти і найняти компанію з необхідним набором навичок для створення вашого проекту на основі ШІ.

У дослідженні Python Developers Survey 2017 (рис. 2.4) ми спостерігаємо, що Python зазвичай використовується для веб-розробки.

What do you use Python for the most? (single answer)

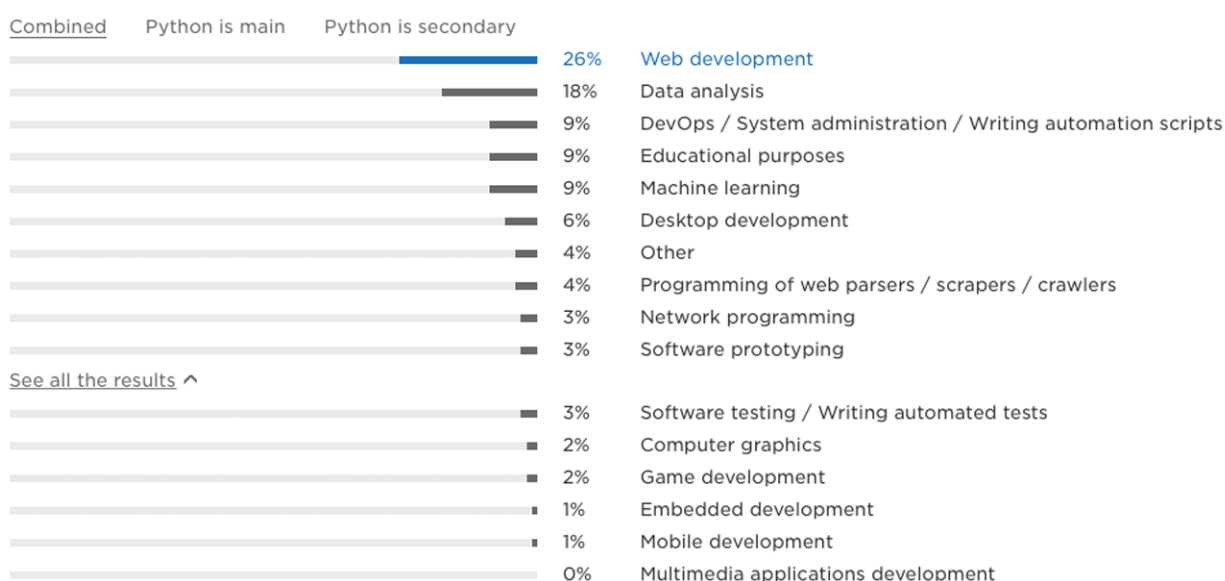


Рис. 2.4 Дослідження Python Developers Survey 2017

На перший погляд, веб-розробка переважає, що становить понад 26% випадків використання, показаних на зображенні нижче. Однак, якщо поєднати науку з даними і машинне навчання, вони становлять приголомшливі 27%.

Інтернет-сховища містять понад 140 тисяч спеціальних програмних пакетів Python. Наукові пакети, такі як Numpy, Scipy і Matplotlib, можуть бути встановлені в програмі, що працює на Python. Ці пакети підходять машинному навчанню і допомагають розробникам виявляти шаблони у великих наборах даних. Python настільки надійний, що Google використовує його для сканування

веб-сторінок, Pixar використовує його для створення фільмів, і Spotify використовує його для рекомендації пісень.

Це загальновідомий факт, що спільнота Python III виросла по всьому світу. Є форуми Python та активний обмін досвідом, пов'язаним з рішеннями для машинного навчання. Для будь-яких завдань, які ви можете мати, шанс досить високий, що хтось інший там має справу з тією ж проблемою. Ви можете знайти поради та рекомендації від розробників. Ви не будете самотніми і обов'язково знайдете найкраще рішення для ваших конкретних потреб, якщо ви звернете до спільноти Python.

2.15 Основні бібліотеки для машинного навчання в Python

Python часто є мовою вибору для розробників, які повинні застосовувати статистичні методи або аналіз даних у своїй роботі. Він також використовується вченими, завдання яких повинні бути інтегровані з веб-додатками або виробничими середовищами.

Python дійсно світиться в області машинного навчання. Його поєднання узгодженого синтаксису, короткого часу розробки та гнучкості робить його придатним для розробки складних моделей і двигунів прогнозування, які можуть підключатися безпосередньо до виробничих систем.

Одним з найбільших активів Python є широкий набір бібліотек.

Бібліотеки - це набори процедур і функцій, написаних на певній мові. Надійний набір бібліотек може полегшити розробникам виконання складних завдань без переписування багатьох рядків коду.

Машинне навчання багато в чому базується на математиці. Зокрема, математична оптимізація, статистика і ймовірність. Бібліотеки Python допомагають дослідникам та математикам, які менш оснащені знаннями розробників, легко «робити машинне навчання».

2.15.1 Scikit-learn для роботи з класичними алгоритмами ML

Scikit-learn є однією з найпопулярніших бібліотек МН. Він підтримує безліч контрольованих і без нагляду алгоритмів навчання. Приклади включають лінійні та логістичні регресії, дерева рішень, кластеризацію, k-засоби тощо.

Вона ґрунтується на двох основних бібліотеках Python: NumPy і SciPy. Він додає набір алгоритмів для звичайних завдань машинного навчання та інтелектуального аналізу даних, включаючи кластеризацію, регресію та класифікацію. Навіть такі завдання, як перетворення даних, вибір ознак і ансамблеві методи, можуть бути реалізовані в декількох рядках.

Для новачка в МН, Scikit-learn є більш ніж достатнім інструментом для роботи, поки не почнете впроваджувати більш складні алгоритми.

2.15.2 Tensorflow для глибинного навчання

Якщо ви перебуваєте у світі машинного навчання, ви, мабуть, чули про, пробували або реалізували якусь форму глибинного алгоритму навчання. Цікаво, що під час написання програми на Python ви можете компілювати і запускати на процесорі або GPU. Тому вам не потрібно писати на рівні C++ або CUDA для запуску на графічних процесорах.

Tensorflow використовує систему багатошарових вузлів, що дозволяє швидко налаштовувати, тренувати та розгортати штучні нейронні мережі з великими наборами даних. Саме це дозволяє Google ідентифікувати об'єкти на фотографіях або розуміти слова, які вони використовують у своєму додатку розпізнавання голосу.

2.15.3 Theano для Deep Learning

Theano - це ще одна хороша бібліотека Python для чисельних обчислень і схожа на NumPy. Theano дозволяє визначити, оптимізувати та оцінити математичні вирази, що включають багатовимірні масиви ефективно.

Те, що відрізняє Theano, полягає в тому, що ця бібліотека користується перевагами GPU. Це дає змогу робити обчислення інтенсивних даних до 100

разів швидше, ніж при роботі на CPU. Швидкість Theano робить її особливо цінною для глибокого навчання та інших складних завдань.

Остаточний випуск бібліотеки Theano минулого року - 2017, версія 1.0.0 з великою кількістю нових можливостей, змін інтерфейсу та поліпшень.

2.15.4 Pandas для вилучення та підготовки даних

Pandas - дуже популярна бібліотека, яка забезпечує високі рівні структур даних, які є простими у використанні та інтуїтивними.

Ця бібліотека має багато вбудованих методів групування, об'єднання даних і фільтрації, а також аналіз часових рядів.

Pandas може легко отримувати дані з різних джерел, таких як бази даних SQL, файли CSV, Excel, JSON і маніпулювати даними для виконання операцій над нею.

2.15.4 Matplotlib для візуалізації даних

Якою б хорошою і складною не була нейронна мережа, вона не буде мати сенсу, якщо ви не можете передавати його іншим людям.

Тут Matplotlib приходить на допомогу. Це стандартна бібліотека Python, яка використовується кожним вченим даних для створення 2D графіків і іншого виду візуалізації даних. Це досить низький рівень, це означає, що він вимагає більше команд для створення красивих графіків і цифр, ніж з деякими розширеними бібліотеками.

Однак зворотний бік цього - гнучкість. Якщо ви маєте достатньо команд, ви можете зробити будь-який графік, який ви хочете використовувати з Matplotlib. Ви можете створювати різноманітні діаграми, від гістограм і графіків розсіювання до недекартових графіків координат.

Він підтримує різні графічні інтерфейси на всіх операційних системах, а також може експортувати графіку у звичайні векторні та графічні формати, такі як PDF, SVG, JPG, PNG, BMP, GIF тощо.

3. Розробка програмного забезпечення

В розробці програмного забезпечення для поставленої задачі можна виділити наступні етапи:

- вибір мови програмування
- вибір бібліотек для машинного навчання
- збір даних та підготовка датасету
- дизайн архітектури моделі
- навчання моделі
- тестування моделі
- створення онлайн сервісу для доступу до моделі в реальному часі 24/7 [Додаток А]

В програмному забезпеченні для нейронної мережі можна виділити наступну структуру [Додаток Б]:

- імпорт усіх необхідних бібліотек та модулів
- завантаження в пам'ять навчального датасету
- завантаження в пам'ять тестового датасету
- розмічення даних (створення масивів з істинними значеннями)
- нормалізація даних (в даному випадку лише приведення до одного розміру зображення)
- дизайн архітектури моделі
- навчання моделі
- тестування моделі (підрахунок метрик оцінки точності)

3.1 Вибір мови програмування

Для даного дослідження основною мовою програмування було обрано Python версії 3.6. На даному етапі розвитку глибокого та й машинного навчання в цілому «флагманом» серед мов програмування є мова Python. Даний вибір був більш детально обґрунтований в пункті 2.14.

3.2 Вибір бібліотек для Python.

Для дизайну архітектури моделі глибинного машинного навчання була обрана відкрита бібліотека Keras, що працювала поверх потужної бібліотеки, розробленою компанією Google, TensorFlow.

Даний вибір обґрунтований тим, що ідеологією даної бібліотеки проста побудова архітектури моделі, стандартизованої загальної структури, але з можливістю гнучкого налаштування під конкретні задачі. А в парі з технічними можливостями TensorFlow дає доступ до більшої кількості налаштувань та можливості правильно працювати з такими компонентами як CPU, GPU та RAM. Також надається доступ до більш розширеної аналітики, що дає можливість більш якісно оцінити працездатність моделі.

Для розробки онлайн сервісу було обрано бібліотеку Flask. Основною її перевагою є підтримка та можливість розробки мікро сервісної архітектури додатків, що в подальшому дає змогу для гнучкого розширення функціоналу. Подібні системи легко масштабуються як в вертикальному, так і в горизонтальному напрямках.

3.3 Підготовка датасету

Однією з основних і найбільш болючих проблем машинного навчання та науки про дані є самі дані, точніше їх відсутність. Незважаючи на те, якими великими кроками рухається вперед технічний прогрес, в усьому світі, зокрема в Україні, досі є петабайти даних, що не оцифровані і зберігаються в архівах, без можливого онлайн доступу до них.

Збір даних є найбільш затратним по часу у машинному навчанні та є актуальною темою досліджень у багатьох спільнотах. Існує в значній мірі дві причини збору даних, які останнім часом стали критичним питанням. По-перше, оскільки машинне навчання стає все більш широко використовуваним, ми бачимо нові можливості використання, які не обов'язково мають достатньо позначених даних. По-друге, на відміну від традиційного машинного навчання, де інженерна особливість є вузьким місцем, глибокі методи навчання

автоматично генерують особливості, але замість цього вимагають великих обсягів позначених даних.

Збір даних дозволяє зафіксувати минулі події, щоб ми могли використовувати аналіз даних для пошуку повторюваних моделей. З цих моделей створюються моделі прогнозування, використовуючи алгоритми машинного навчання, які шукають тенденції та передбачають майбутні зміни.

Моделі прогнозування є настільки ефективними, наскільки і дані, на основі вони побудовані, тому хороші практики збору даних мають вирішальне значення для розробки високоефективних моделей. Дані повинні бути без помилок і містити відповідну інформацію для поточного завдання. Наприклад, модель кредитного дефолту не зміниться від розмірів популяції тигрів, але з часом може змінитись від цін на газ.

Для даного дослідження було використано заздалегідь підготований датасет (License: CC BY 4.0), що складався з 5 863 рентгенівських знімків грудної клітини дітей. Датасет включає в себе 4266 знімки з пневмонією, з яких 2 538 бактеріальна пневмонія, а 1 728 вірусна, а також 1 576 знімків здорових легень (рис. 3.1-3.4) [8].

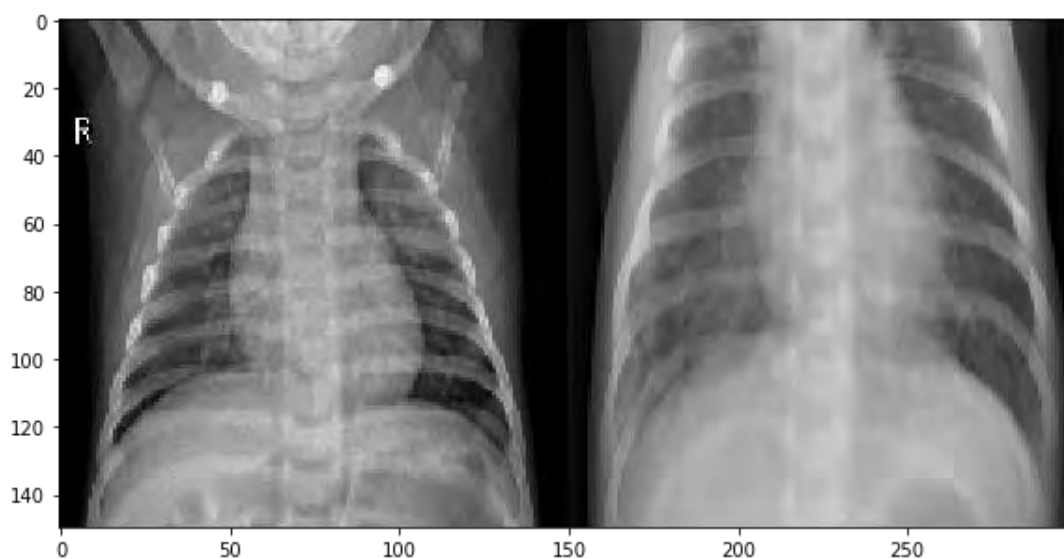


Рис. 3.1 Зліва здорові легені, справа бактеріальна пневмонія

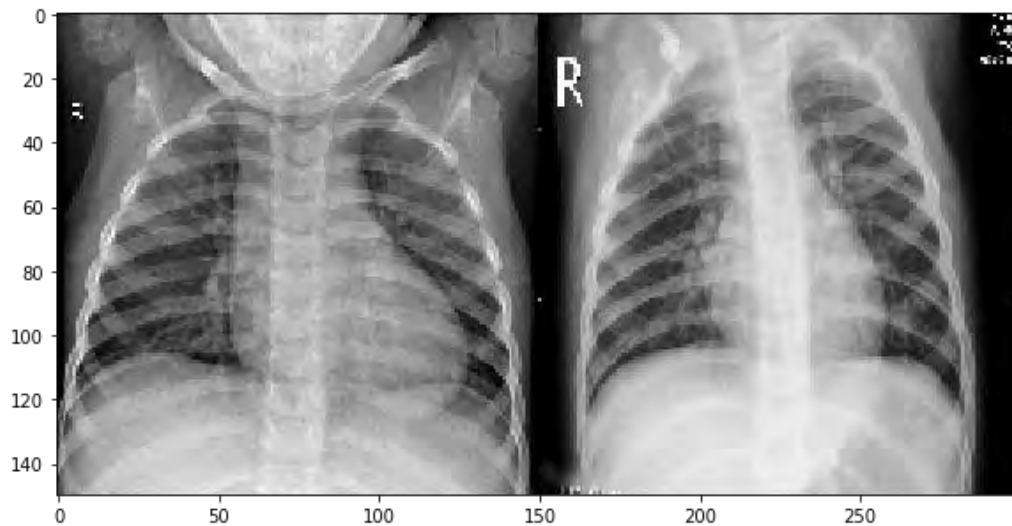


Рис. 3.2 Зліва здорові легені, справа бактеріальна пневмонія

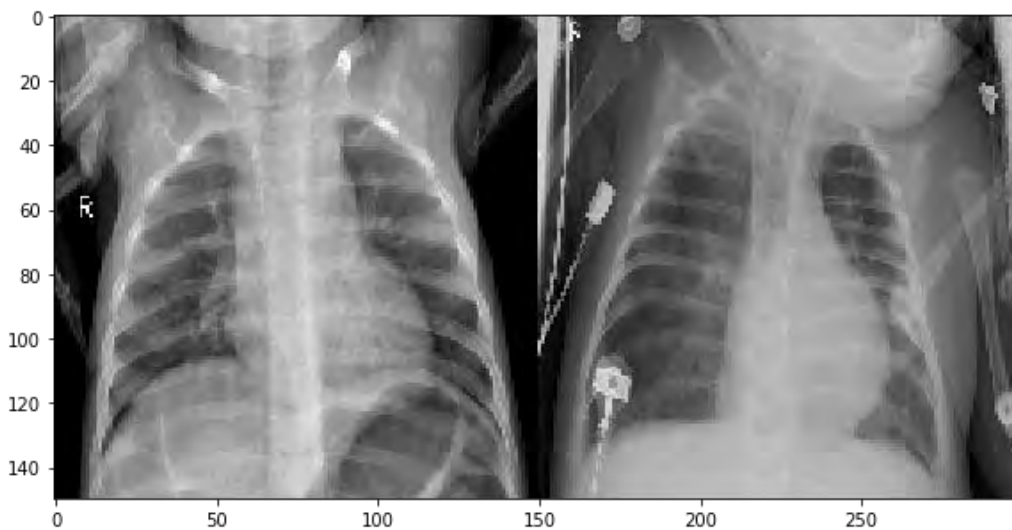


Рис. 3.3 Зліва здорові легені, справа вірусна пневмонія

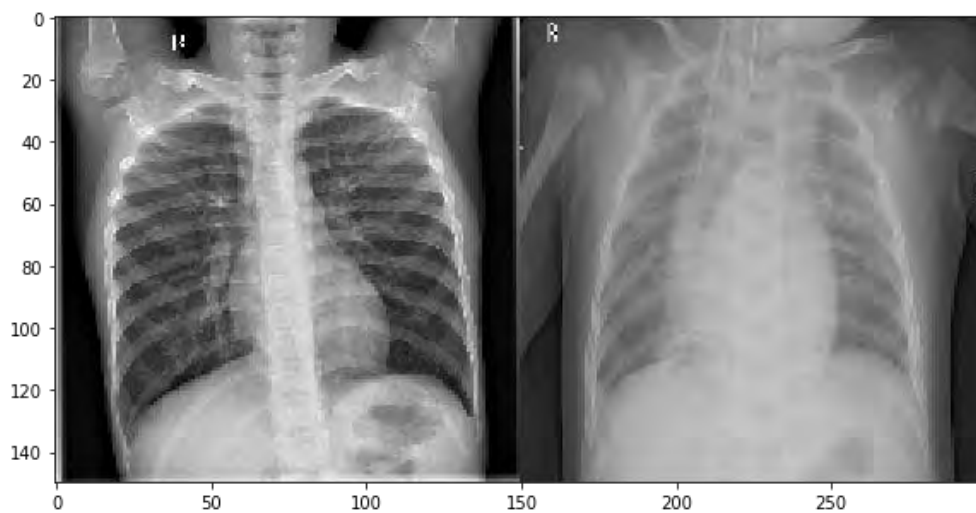


Рис. 3.4 Зліва здорові легені, справа бактеріальна пневмонія

Критерієм відбору розмічених знімків була верифікація діагнозу двома незалежними кваліфікованими лікарями. Якщо ж діагнози лікарів збігались з реальною проблемою чи її відсутністю у пацієнта, знімок включався до загального датасету з відповідною міткою. Виходячи з загального положення та опираючись на сукупність всіх вищевказаних та згаданих фактів можна дійти висновку, що даному датасету можна довіряти та можна проводити аналіз на його основі.

Як ми можемо спостерігати пневмонію в більшості випадках досить добре видно на знімку, проте не всі випадки пневмонії є достатньо видимі для неозброєного людського ока і досить важко робити будь-які припущення. Тому проблема машинного навчання постає досить гостро, адже це повноцінний ще один етап верифікації, в процесі якого можна поміти ті захворювання та проблеми, які досить легко пропустити лікарю.

Для коректної подальшої оцінки моделі потрібно розділити датасет на дві частини. Частина знімків, а саме 90%, буде розподілена до навчального датасету, інші ж 10% будуть розподілені до тестового датасету. Дана операція необхідна для того, щоб мати можливість протестувати модель на знімках, які вона ще «не знає».

3.4 Дизайн архітектури моделі

Для розробки було обрано sequential або ж KNF модель, що дає змогу зручно будувати багат шарові моделі. З детальним описом шарів та їх специфікацією можна ознайомитись в офіційній документації [9]. Емпіричним шляхом було з'ясовано, що саме ця комбінація шарів (рис. 3.5) дає найкращий результат в обробці зображень. В даному випадку 19 шарів.

Layer (type)	Output Shape	Param #
conv2d_97 (Conv2D)	(None, 16, 150, 150)	448
conv2d_98 (Conv2D)	(None, 16, 150, 150)	2320
max_pooling2d_47 (MaxPooling)	(None, 16, 75, 75)	0
conv2d_99 (Conv2D)	(None, 32, 75, 75)	4640
conv2d_100 (Conv2D)	(None, 32, 75, 75)	9248
max_pooling2d_48 (MaxPooling)	(None, 32, 37, 37)	0
conv2d_101 (Conv2D)	(None, 64, 37, 37)	18496
conv2d_102 (Conv2D)	(None, 64, 37, 37)	36928
max_pooling2d_49 (MaxPooling)	(None, 64, 18, 18)	0
conv2d_103 (Conv2D)	(None, 96, 18, 18)	55392
conv2d_104 (Conv2D)	(None, 96, 16, 16)	83040
max_pooling2d_50 (MaxPooling)	(None, 96, 8, 8)	0
conv2d_105 (Conv2D)	(None, 128, 8, 8)	110720
conv2d_106 (Conv2D)	(None, 128, 6, 6)	147584
max_pooling2d_51 (MaxPooling)	(None, 128, 3, 3)	0
flatten_11 (Flatten)	(None, 1152)	0
dense_21 (Dense)	(None, 64)	73792
dropout_11 (Dropout)	(None, 64)	0
dense_22 (Dense)	(None, 2)	130
=====		
Total params: 542,738		
Trainable params: 542,738		
Non-trainable params: 0		
=====		
None		

Рис. 3.5 Структура моделі

3.5 Навчання моделі

Навчання моделі проходило у 8 ітерацій, так званих епох. Тести показали, що після 8 епохи продовжувати навчання немає сенсу, адже точність припиняє збільшуватись, а функція втрат спадати, таке навчання може призвести до перенавчання моделі (рис. 3.6).

На робочій станції наступної конфігурації:

- Intel i7-3300K
- Nvidia GTX740m
- 12 GB RAM
- 240 Gb SSD

Навчання зайняло 8,5 годин.

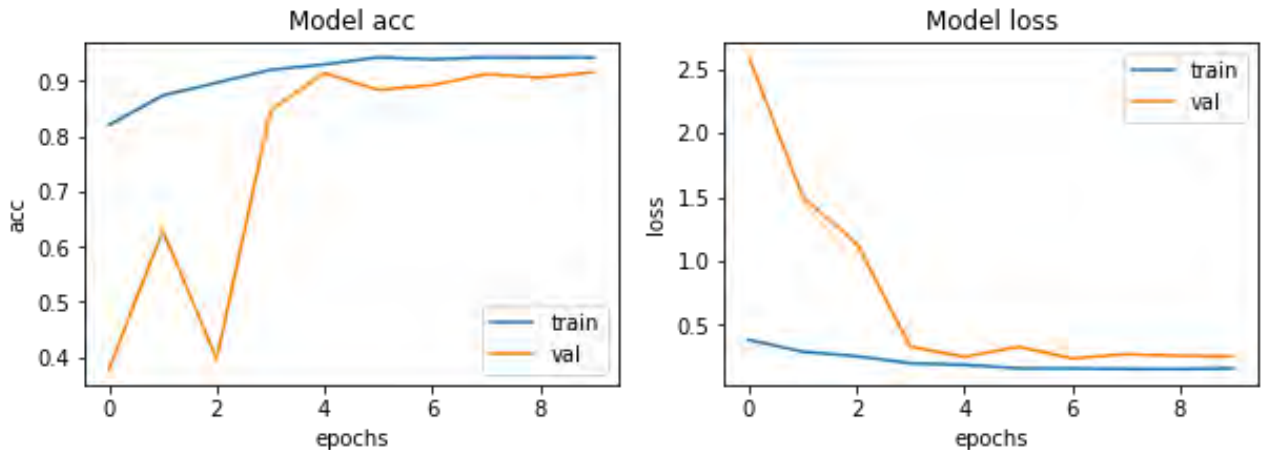


Рис. 3.6 Графіки оцінки точності та функції втрат

На вище наведених графіках можна чітко побачити, що зі збільшенням кількості епох точність перестає збільшуватись, а функція втрат перестає зменшуватись. Отже 8 епох це оптимальна кількість для навчання нашої моделі, адже збільшення кількості епох не завжди приводить до збільшення точності.

3.6 Тестування моделі

Для тестування моделі використовувався заздалегідь підготований датасет, який не брав участі у навчанні моделі.

Для тестування точності моделі зазвичай використовуються чотири метрики оцінки точності: accuracy, precision, recall, F1. Кожна з них характеризує певну точність, отриману в результаті обчислень, тому вони всі можуть бути різними. Для того щоб обрахувати метрики потрібно спершу отримати confusion matrix (рис 3.7).

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Рис. 3.7 Таблиця формування confusion matrix

І на основі цієї матриці можна обчислювати всі матрики.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

$$Precision = \frac{TP}{TP+FP} \quad (3.2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.3)$$

де TP – TruePositive, TN – TrueNegative, FP – FalsePositive, FN – False Negative

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (3.4)$$

На тестовому датасеті модель показала наступні результати (рис. 3.8).

```

CONFUSION MATRIX -----
[[191  43]
 [ 13 377]]

TEST METRICS -----
Accuracy: 91.02564102564102%
Precision: 89.76190476190476%
Recall: 96.66666666666667%
F1-score: 93.08641975308642

TRAIN METRIC -----
Train acc: 94.23

```

Рис. 3.8 Основні метрики оцінки точності

Як ми можемо бачити з наведених вище метрик оцінки точності дана модель показує достатньо точний результат та велику вірогідність правильного

діагнозу. Отже, дану модель можна використовувати як допоміжний засіб верифікації діагнозу.

3.7 Розробка онлайн сервісу

Для зручного доступу користувачів до моделі, незалежно від рівня освіченості в програмуванні, було прийнято рішення розробити простий та зручний в користуванні онлайн сервіс для швидкого доступу до моделі в реальному часі.

4. Результати дослідження

Результатом даного дослідження є сервіс, що в режимі онлайн, з достатньо великою точністю (рис. 3.8), надає результат розпізнавання рентген знімку.

Загальна структура роботи сервісу виглядає наступними чином:

1. Завантаження головного екрану сервісу (рис. 4.1)

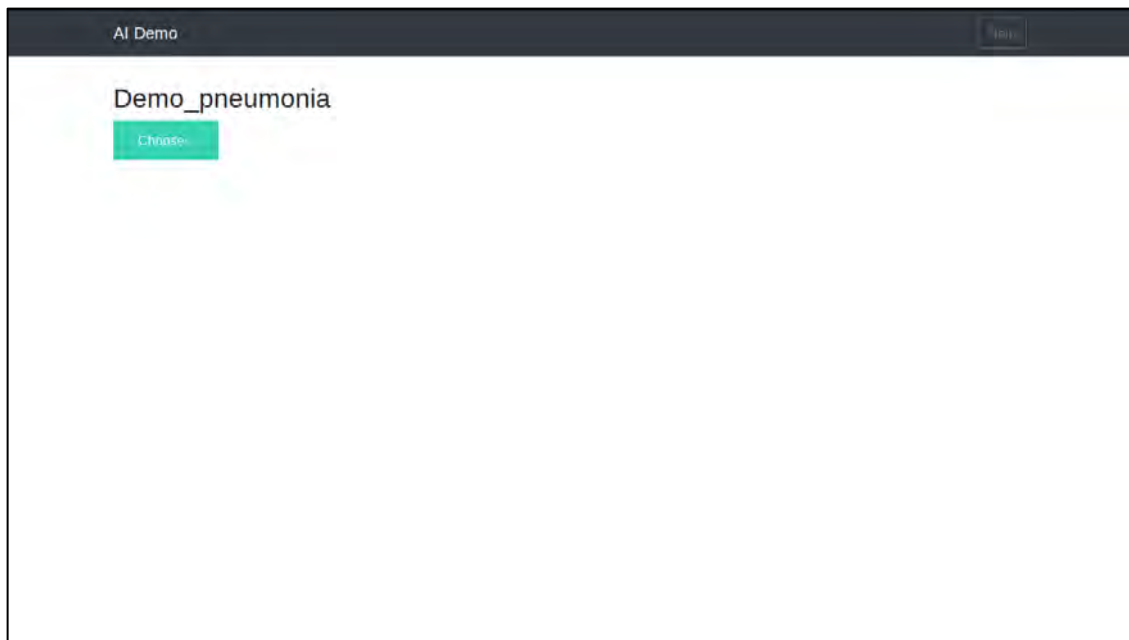


Рис 4.1. Головний екран сервісу

2. Завантаження користувачем рентген знімку (рис. 4.2).

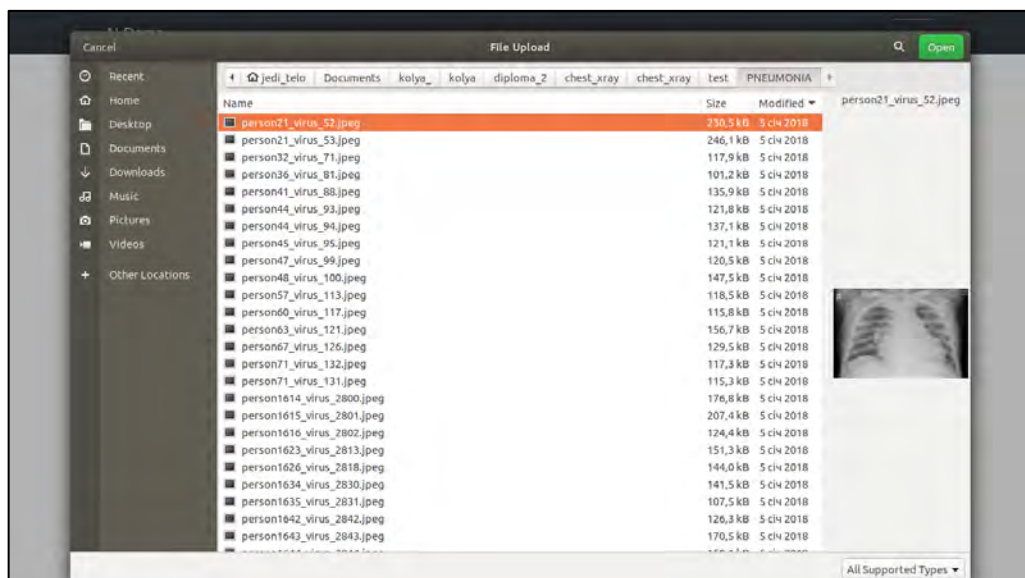


Рис. 4.2 Завантаження знімку

3. Розроблена вище, модель обробляє даний знімок (рис. 4.3).

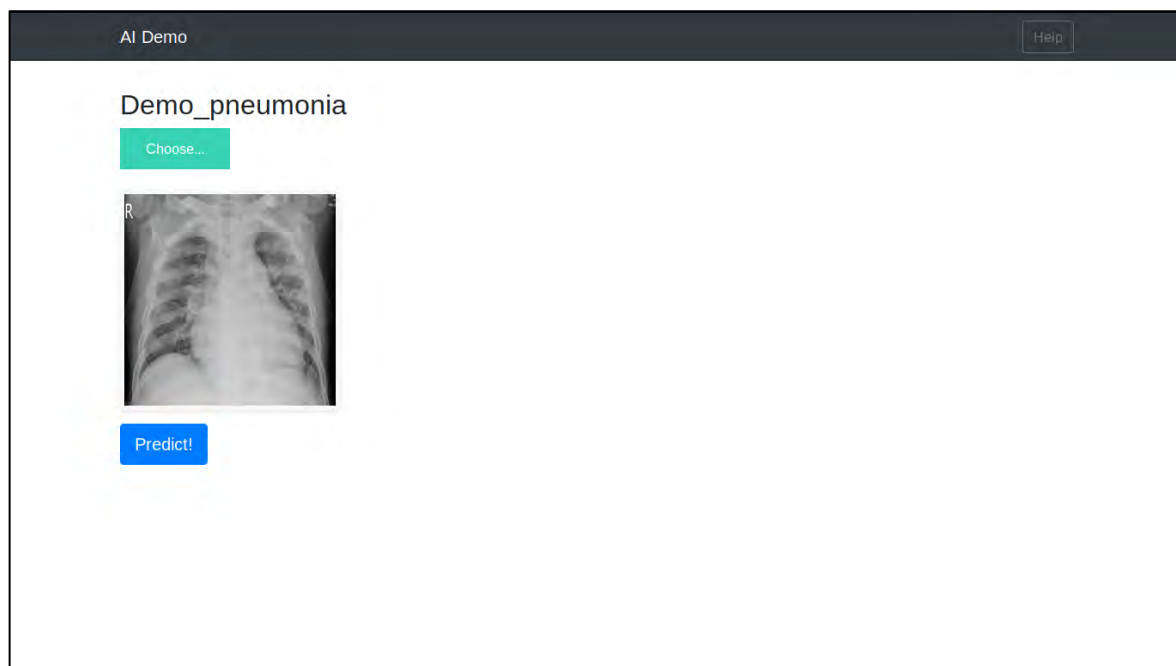


Рис. 4.3 Обробка знімку

4. Користувачеві надсилається відповідь з діагнозом в зручному для нього вигляді (рис. 4.4).

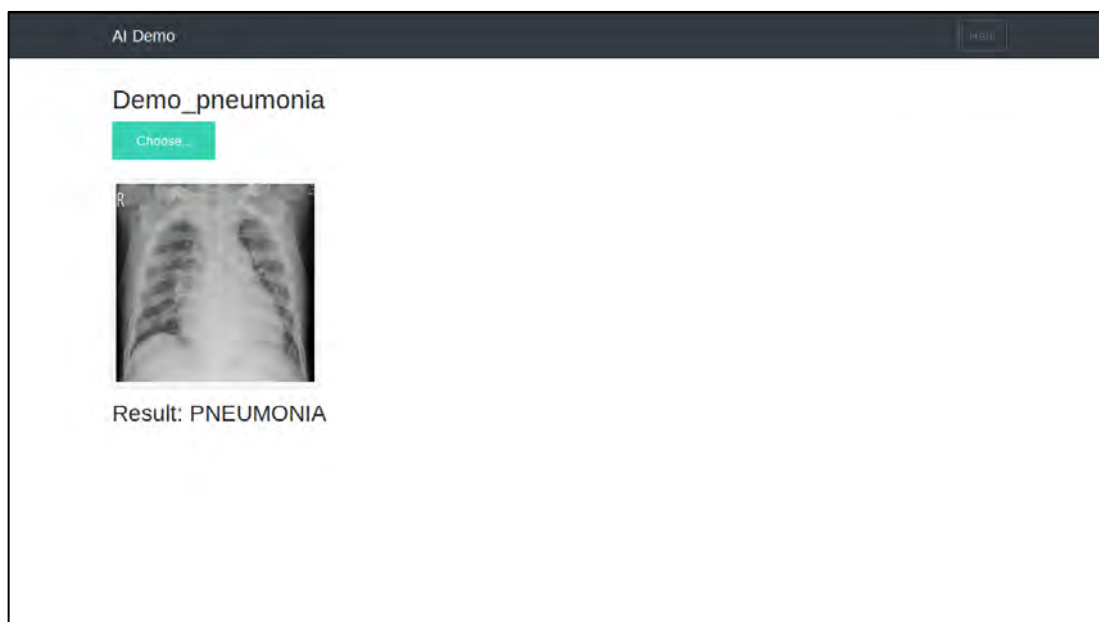


Рис. 4.4 Результат роботи сервісу

Даний сервіс доступний онлайн, з будь-якого девайсу, на якому є доступ в інтернет

Користувач, незалежно від того, лікар це чи пацієнт, може отримати ще одну кваліфіковану точку зору, ще одне підтвердження, або ж спростування діагнозу, маючи лише цифрове зображення рентген знімку.

В даному дослідженні система розпізнає лише вірусну та бактеріальну пневмонію. Але основною перевагою такого підходу до діагностування є те, що за наявності необхідної кількості даних, систему можна масштабувати. Це, в свою чергу, означає, що перед нами є можливість запровадити машинне навчання майже в усі галузі медицини, що має позитивно відобразитись на статистиці хибних діагнозів.

Ще однією перевагою даного методу є те, що система має можливість постійно «навчатись» та підвищувати точність розпізнавання. Це стане можливим, коли користувачі почнуть користуватись даним сервісом і завантажувати нові знімки, які в подальшому можна використовувати для навчання моделі.

Висновки

Без сумнівів глибинне машинне навчання в сучасному світі надбирає популярності в різних сферах життєдіяльності, зокрема медицині. Хоча це досить молодий вид діагностики, проте він вже зарекомендував себе як один з високоточних, швидких та зручних в користуванні методів. Проте на даному етапі розвитку все ще не може повноцінно виключити людину з етапу діагностики, однією з причин є мала кількість даних.

В роботі розглянуто основні засоби розробки систем машинного навчання та одна з можливих сфер застосування нейронних мереж в медицині, а саме – рентгенологія.

На протязі роботи був розроблений онлайн сервіс для діагностування захворювань легень за рентгенівськими знімками. В якості тестової хвороби була обрана пневмонія.

Система показала досить велику достовірність, попри достатньо невелику кількість вхідних даних. Даний сервіс має ряд вагомих переваг над звичними методами:

- висока точність
- доступ в режимі онлайн 24/7
- немає необхідності в медичній освіті
- відсутність людського фактору
- короткий час поставлення діагнозу (обмежується швидкістю інтернет з'єднання)
- висока продуктивність

Хоча в сучасному світі більшість все ще надають перевагу традиційному методу діагностування, заперечувати ефективність та переваги машинного навчання неможливо і на даному етапі, хоча б в якості допоміжного способу діагностування, варто використовувати дану методологію.

Список використаної літератури

1. Galagan R.M. Analysis of application of neural networks to improve the reliability of active thermal NDT / R. M. Galagan, A. S. Momot // KPI Science News. – Kyiv. – 2019. – № 1 (2019). – P. 7-14 DOI: <https://doi.org/10.20535/kpi-sn.2019.1.157374>
2. R. Galagan and A. Momot, "The Use of Backpropagation Artificial Neural Networks in Thermal Tomography", *2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC)*, Kyiv, Ukraine, 2018, pp. 1-6. (doi: 10.1109/SAIC.2018.8516880)
3. Fischbach FT, Dunning MB III, eds. Manual of Laboratory and Diagnostic Tests, 8th ed. Philadelphia: Lippincott Williams and Wilkins, 2009.
4. Pagana KD, Pagana TJ. Mosby's Manual of Diagnostic and Laboratory Tests, 4th ed. St. Louis: Mosby Elsevier, 2010.
5. Свіридов О. І. Анатомія людини. — Київ : Вища школа, 2001. — 240 с.
6. Галаган Р.М. Аналіз методів цифрової обробки термограм / Р.М. Галаган, А.С. Момот // Вісник Національного Технічного Університету України «КПІ». Серія приладобудування. – Київ. – 2018. – № 55(1). – С. 108-117.
7. Бурау Н.І. Синтез нейронної мережі для багатоканальної діагностики елементів конструкції в експлуатації / Н.І. Бурау, А.Г. Протасов, П.С. Мироненко, С.С. Рупіч. – Methods and Devices of Quality Control. – 2015. 2(35). – С. 83-93
8. Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, v2: [Електрон. ресурс]. - <https://data.mendeley.com/datasets/rsbjbr9sj/2> (DOI <http://dx.doi.org/10.17632/rsbjbr9sj.2>)
9. Keras: The Python Deep Learning library: [Електрон. ресурс]. - <https://keras.io/>

Додатки

Технічне завдання

1. Найменування системи.

„Онлайн-сервіс для діагностування захворювань легень за рентгенівськими знімками”.

2. Мета виконання дипломної роботи та призначення системи.

Мета роботи полягає в дослідженні високоточного методу діагностики захворювань легень на основі рентген знімків. Його розробка, розробка онлайн інтерфейсу для зручного доступу в режимі реального часу та складання документації до нього.

3. Передумови для розробки приладу.

Дана розробка виконується відповідно до тематики дипломної роботи, затвердженої кафедрою приладів та систем неруйнівного контролю приладобудівного факультету НТУУ „КПІ”.

4. Вимоги до технічних характеристик.

Програмна частина повинна забезпечувати можливість виконання перелічених нижче функцій:

- Онлайн доступ до сервісу 24/7
- Можливість завантажити рентген знімок формату .jpeg (.jpg)
- Попередній перегляд завантаженого знімку
- Опрацювання знімку за допомогою нейронної мережі
- Подання результату діагностики в зручному для користувача вигляді

5. Робочі умови експлуатації приладу.

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації

6. Вимоги до забезпечення надійного функціонування програмного забезпечення.

Надійне (стійке) функціонування програмного забезпечення повинно бути забезпечено виконанням замовником сукупності організаційно-технічних заходів, перелік яких наведено нижче:

- організацією безперебійного живлення технічних засобів;
- використанням ліцензійного програмного забезпечення;

7. Вимоги до програмного забезпечення.

Основний сервер системи. Сервер встановлюється в приміщенні і має постійну білу виділену IP адресу в зовнішній мережі. Точне місце установки визначається і узгоджується із замовником в процесі розробки системи. Сервер будується на базі IBM-сумісного комп'ютера, встановленого в серверну стійку і включає в себе:

- оперативну пам'ять об'ємом, 16 Гігабайт, не менше;
- 1 SSD, 512 Мбайт, не менше;
- операційну систему Ubuntu LTS;
- програмне середовище Python.

8. Спеціальні вимоги

Система повинна забезпечувати одночасну роботу користувачів за допомогою веб інтерфейсу

9. Стандартизація та уніфікація.

До системи повинна бути прикладена експлуатаційна документація відповідно ДСТУ 25565-94.

10. Вимоги до техніки безпеки.

10.1. Вимоги до безпеки конструкції, електробезпеки керування та обслуговування системи повинні відповідати ДСТ 122007-75 та ДСТУ 123019-80.

10.2. Час відновлення після відмови, викликаного збоєм електроживлення технічних засобів (іншими зовнішніми факторами), які не є фатальним збоєм (не крах) операційної системи, не повинно перевищувати 30-ти хвилин за умови дотримання умов експлуатації технічних і програмних засобів.

Час відновлення після відмови, викликаного несправністю технічних засобів, фатальним збоєм (крахом) операційної системи, не повинно перевищувати часу, необхідного на усунення несправностей технічних засобів і перевстановлення програмних засобів.

Додаток А

```
from future import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
import cv2
import skimage
from random import shuffle
from tqdm import tqdm
import scipy
from skimage.transform import resize

# Keras
from keras.applications.imagenet_utils import preprocess_input, decode_predictions
from keras.models import load_model
from keras.preprocessing import image
from keras import backend as K
K.set_image_dim_ordering('th')

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

# Define a flask app
app = Flask(__name__)

# Model saved with Keras model.save()
MODEL_PATH = 'models/weights.hdf5'

# Load your trained model
def swish_activation(x):
    return (K.sigmoid(x) * x)

model = load_model(MODEL_PATH, custom_objects={'swish_activation':
swish_activation})
model._make_predict_function() # Necessary
print('Model loaded. Start serving...')
```

```

def model_predict(img_path, model):
    imgs = []
    img = cv2.imread(img_path)
    #img = image.img_to_array(img)
    img = skimage.transform.resize(img, (150, 150, 3))
    img = np.asarray(img)
    imgs.append(img)
    imgs = np.asarray(imgs)
    imgs = imgs.reshape(1,3,150,150)
    preds = model.predict(imgs)
    prob = model.predict_proba(imgs)
    class_ = model.predict_classes(imgs)
    return preds, prob, class_

@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uphttp://localhost:5000/loads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)

        # Make prediction
        preds, prob, class_ = model_predict(file_path, model)

        # Process your result for human
        # pred_class = preds.argmax(axis=-1) # Simple argmax
        #pred = np.argmax(preds,axis = 0)
        #pred_class = decode_predictions(preds, top=1) # ImageNet Decode
        if class_[0] == 0:
            result = "Normal"
        else:

```

```
result = "PNEUMONIA"
```

```
#result = str(prob)
```

```
#result = 'preds: ' + str(preds) + ' proba: ' + str(prob) + ' class: ' + str(class_) # Convert  
to string
```

```
return result
```

```
return None
```

```
if name == '__main__':
```

```
#app.run(port=5002, debug=True)
```

```
# Serve the app with gevent
```

```
http_server = WSGIServer("", 5000), app)
```

```
http_server.serve_forever()
```

Додаток Б

```
.. code:: ipython3
```

```
import pandas as pd
import cv2
import numpy as np
import os
from random import shuffle
from tqdm import tqdm
import scipy
import skimage
from skimage.transform import resize
```

```
.. code:: ipython3
```

```
print(os.listdir("./chest_xray/test/"))
```

```
.. code:: ipython3
```

```
TRAIN_DIR = "./chest_xray/train/"
TEST_DIR = "./chest_xray/test/"
test = "./chest_xray/test/a/"
```

```
.. code:: ipython3
```

```
def get_label(Dir):
    for nextdir in os.listdir(Dir):
        if not nextdir.startswith('.'):
            if nextdir in ['NORMAL']:
                label = 0
            elif nextdir in ['PNEUMONIA']:
                label = 1
            else:
                label = 2
    return nextdir, label
```

```
.. code:: ipython3
```

```
def preprocessing_data(Dir):
    X = []
    y = []

    for nextdir in os.listdir(Dir):
        nextdir, label = get_label(Dir)
```

```

temp = Dir + nextdir

for image_filename in tqdm(os.listdir(temp)):
    path = os.path.join(temp + '/' , image_filename)
    img = cv2.imread(path,cv2.IMREAD_GRAYSCALE)
    if img is not None:
        img = skimage.transform.resize(img, (150, 150, 3))
        img = np.asarray(img)
        X.append(img)
        y.append(label)

X = np.asarray(X)
y = np.asarray(y)

return X,y

.. code:: ipython3

def get_data(Dir):
    X = []
    y = []
    for nextDir in os.listdir(Dir):
        if not nextDir.startswith('.'):
            if nextDir in ['NORMAL']:
                label = 0
            elif nextDir in ['PNEUMONIA']:
                label = 1
            else:
                label = 2

    temp = Dir + nextDir

    for file in tqdm(os.listdir(temp)):
        img = cv2.imread(temp + '/' + file)
        if img is not None:
            img = skimage.transform.resize(img, (150, 150, 3))
            #img_file = scipy.misc.imresize(arr=img_file, size=(150, 150, 3))
            img = np.asarray(img)
            X.append(img)
            y.append(label)

    X = np.asarray(X)
    y = np.asarray(y)
    return X,y

```

```

.. code:: ipython3

X_train, y_train = get_data(TRAIN_DIR)

.. code:: ipython3

X_test , y_test = get_data(TEST_DIR)

.. code:: ipython3

print(X_train.shape,\n',X_test.shape)

.. code:: ipython3

print(y_train.shape,\n',y_test.shape)

.. code:: ipython3

from keras.utils.np_utils import to_categorical

y_train = to_categorical(y_train, 2)
y_test = to_categorical(y_test, 2)

.. code:: ipython3

y_t = to_categorical(y_t, 2)

.. code:: ipython3

print(y_train.shape,\n',y_test.shape)

.. code:: ipython3

Pimages = os.listdir(TRAIN_DIR + "PNEUMONIA")
Nimages = os.listdir(TRAIN_DIR + "NORMAL")

.. code:: ipython3

import matplotlib.pyplot as plt
def plotter(i):
imagep1 = cv2.imread(TRAIN_DIR+"PNEUMONIA/"+Pimages[i])
imagep1 = skimage.transform.resize(imagep1, (150, 150, 3) , mode = 'reflect')
imagen1 = cv2.imread(TRAIN_DIR+"NORMAL/"+Nimages[i])
imagen1 = skimage.transform.resize(imagen1, (150, 150, 3))
pair = np.concatenate((imagen1, imagep1), axis=1)

```

```

print("(Left) - No Pneumonia Vs (Right) - Pneumonia")
print("-----")
print("-----")
plt.figure(figsize=(10,5))
plt.imshow(pair)
plt.show()
for i in range(0,5):
    plotter(i)

.. code:: ipython3

import seaborn as sns
count = y_train.sum(axis = 0)
sns.countplot(x = count)

.. code:: ipython3

from keras.callbacks import ReduceLRonP
lateau , ModelCheckpoint
lr_reduce = ReduceLRonPlateau(monitor='val_acc', factor=0.1, epsilon=0.0001,
patience=1, verbose=1)

.. code:: ipython3

filepath="weights.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1,
save_best_only=True, mode='max')

.. code:: ipython3

from keras.models import Sequential
from keras.layers import Dense , Activation
from keras.layers import Dropout
from keras.layers import Flatten
from keras.constraints import maxnorm
from keras.optimizers import SGD , RMSprop
from keras.layers import Conv2D , BatchNormalization
from keras.layers import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
K.set_image_dim_ordering('th')
from sklearn.model_selection import GridSearchCV

```

```
from keras.wrappers.scikit_learn import KerasClassifier
```

```
.. code:: ipython3
```

```
X_train=X_train.reshape(5216,3,150,150)
```

```
X_test=X_test.reshape(624,3,150,150)
```

```
.. code:: ipython3
```

```
X_t=X_t.reshape(2,3,150,150)
```

```
.. code:: ipython3
```

```
from keras.utils.generic_utils import get_custom_objects
```

```
def swish_activation(x):
```

```
return (K.sigmoid(x) * x)
```

```
model = Sequential()
```

```
model.add(Conv2D(16, (3, 3), activation='relu', padding="same",  
input_shape=(3,150,150)))
```

```
model.add(Conv2D(16, (3, 3), padding="same", activation='relu'))
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(32, (3, 3), activation='relu', padding="same",  
input_shape=(3,150,150)))
```

```
model.add(Conv2D(32, (3, 3), padding="same", activation='relu'))
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(64, (3, 3), activation='relu', padding="same"))
```

```
model.add(Conv2D(64, (3, 3), padding="same", activation='relu'))
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(96, (3, 3), dilation_rate=(2, 2), activation='relu',  
padding="same"))
```

```
model.add(Conv2D(96, (3, 3), padding="valid", activation='relu'))
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(128, (3, 3), dilation_rate=(2, 2), activation='relu',
```



```

padding="same"))
model.add(Conv2D(128, (3, 3), padding="valid", activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(64, activation=swish_activation))
model.add(Dropout(0.4))
model.add(Dense(2, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
optimizer=RMSprop(lr=0.00005),
metrics=['accuracy'])

print(model.summary())

.. code:: ipython3

batch_size = 256
epochs = 8

.. code:: ipython3

history = model.fit(X_train, y_train, validation_data = (X_test, y_test)
, callbacks=[lr_reduce, checkpoint],
epochs=epochs)

.. code:: ipython3

import matplotlib.pyplot as plt

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

```

```
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
.. code:: ipython3
```

```
from sklearn.metrics import confusion_matrix
pred = model.predict(X_test)
pred = np.argmax(pred,axis = 1)
y_true = np.argmax(y_test,axis = 1)
```

```
.. code:: ipython3
```

```
prob = model.predict_proba(X_test)
class_ = model.predict_classes(X_test)
```

```
.. code:: ipython3
```

```
CM = confusion_matrix(y_true, pred)
from mlxtend.plotting import plot_confusion_matrix
fig, ax = plot_confusion_matrix(conf_mat=CM , figsize=(5, 5))
plt.show()
```