

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Нейромережева модель аналізу дитячих малюнків для
діагностики емоційного стану дитини

Виконав:

студент 2 курсу, групи КІТм-21-2

Дикий С. А.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні інтелектуальні
технології

Керівник проф. Корабльов М.М.

Допускається до захисту

(підпис)

Зав. кафедри

(підпис)

проф. Руденко О.Г.

2022 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ ____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Дикому Станіславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Нейромережева модель аналізу дитячих малюнків для діагностики емоційного стану дитини

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

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

3. Вхідні дані до роботи _____

1) виявлення характерних ознак емоційного стану на дитячих малюнках;

2) побудова моделі нейронної мережі для діагностики емоційного стану дитини;

3) середовище моделювання – Rucharm;

4) мова програмування – Python.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) огляд предметної області;

2) аналіз предмету дослідження;

3) дослідження нейронних мереж;

4) дослідження характерних ознак емоційного стану за малюнками;

5) розробка нейромережевої моделі аналізу дитячих малюнків;

6) експериментальні дослідження;

7) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайди - 17 арк. ф. А4

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача та узгодження теми проєкту	07.09.2022	1
2	Огляд стану проблеми та постановка задачі	09.09-14.09	2
3	Аналіз літератури за напрямком магістерської роботи	14.09-21.09	3
4	Аналіз методів діагностики емоційного стану за малюнками	21.09-28.09	4
5	Аналіз існуючих нейронних мереж	21.09-28.09	5
6	Розробка тестових нейронних мереж	28.09-12.10	6
7	Експериментальні дослідження	12.10-02.11	7
8	Підготовка графічного матеріалу	23.11-07.12	8
9	Перевірка виконаного проєкту керівником	10.12.2022	9
10	Захист проєкту	21.12.2022	10

Дата видачі завдання 01 вересня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Корабльов М.М
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка магістерської кваліфікаційної роботи: 102 с., 55 рис., 17 джерел.

МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ШТУЧНИЙ ІНТЕЛЕКТ, ЕМОЦІЙНИЙ СТАН, ДАТАСЕТ, ЗАДАЧА КЛАСИФІКАЦІЇ

Ціль роботи – створення нейромережевої моделі аналізу дитячих малюнків для діагностики емоційного стану дитини, яка може частково допомогти фахівцю у повсякденному житті виконувати частину його роботи, а також провести первинний аналіз емоційного стану дитини онлайн, без участі спеціалістів.

Об'єкт дослідження – ознаки та закономірності дитячих малюнків, за якими можна зробити висновок про емоційне здоров'я дитини.

Предмет дослідження – нейромережеві моделі та алгоритми, які можуть бути використані для виявлення характерних ознак на дитячих малюнках, що дасть змогу зробити висновки про емоційний та психологічний стан дитини.

Методи дослідження – нейронні мережі та машинні алгоритми.

Таким чином в ході роботи була побудована нейромережева модель аналізу дитячих малюнків для діагностики емоційного стану дитини. Дана система представлена як консольний додаток. Дана система показала, що вона може виконувати поставлені завдання.

ABSTRACT

Explanatory note of master's qualification work: 102 pages, 55 figures, 17 sources.

MACHINE LEARNING, NEURAL NETWORKS, CONVOLUTIONAL NEURAL NETWORKS, ARTIFICIAL INTELLIGENCE, EMOTIONAL CONDITION, DATASET, CLASSIFICATION

The goal of the work is creation of a neural network model for the analysis of children's drawings for the diagnosis of the child's emotional state, which can partially help the specialist in everyday life to perform part of his work, as well as conduct a primary analysis of the child's emotional state online, without the participation of specialists.

The object of research signs and regularities of children's drawings, which can be used to draw a conclusion about the child's emotional health.

The subject of research is neural network models and algorithms that can be used to identify characteristic features in children's drawings, which will make it possible to draw conclusions about the child's emotional and psychological state.

Research methods – neural networks and machine algorithms.

Thus, in the course of the work, a neural network model of the analysis of children's drawings was built to diagnose the child's emotional state. This system is presented as a console application. This system has shown that it can perform the assigned tasks.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ Комп'ютерних інтелектуальних технологій та систем _____

АНОТАЦІЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти _____ другий (магістерський)

Нейромережева модель аналізу дитячих малюнків для
діагностики емоційного стану дитини

Виконав:

студент 2 курсу, групи КІТм-21-2

Дикий С. А.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні
інтелектуальні технології

Керівник проф. Корабльов М.М.

2022 р.

АНОТАЦІЯ

Дикий С.А. Нейромережева модель аналізу дитячих малюнків для діагностики емоційного стану дитини — Магістерська кваліфікаційна робота.

Актуальність теми дослідження. У сучасному світі безперервного споживання інформації, стресу та психологічного тиску дуже гостро стоїть питання емоційного та психологічного здоров'я людей, особливо дітей. Проте не завжди є легкою задачею виявити можливі проблеми, переживання дітей та вчасно звернутися до фахівців. Виявлення дитячих емоцій за допомогою їхніх малюнків є старою технікою, яку часто використовують психологи. Терапевти використовують різні техніки за допомогою малюнків, щоб оцінити емоції та зрілість дітей. Одним із найпопулярніших є тест «Сімейний малюнок». Тест «Сімейний малюнок» дає зрозуміти, як дитина сприймає свої найближчі стосунки. Це простий спосіб зрозуміти якість стосунків і спілкування, а також те, як діти будують свою реальність на основі сімейних стосунків. Ця техніка існує вже понад шість десятиліть. Психіатр Майлз Порот створив його в 1951 році, і зараз це одна з найпоширеніших методик, які використовуються для оцінки особистості дітей віком від 5 до 16 років. Будь-які тести та діагностики подібного плану базуються на знаходженні та виявленні певних закономірностей. Саме тому стає актуальним питання автоматизації цього процесу. Адже головна мета та перевага будь-якої автоматизації, зокрема автоматизації з використанням машинного навчання та нейронних мереж, це покращення та пришвидшення процесів. Отже, використання нейронних мереж для задачі аналізу емоційного стану дитини з одного боку виводить з рівняння людський фактор, що позитивно впливає на правильний результат, та з іншого — виводить питання діагностики емоційного стану на новий рівень, адже дає можливість без втручання спеціалістів, сидячи вдома перед комп'ютером, провести первинну діагностику та виявити потенційну проблему на

ранньому етапі.

У магістерській роботі досліджено науково-прикладну проблему створення нейромережевої моделі аналізу дитячих малюнків для діагностики емоційного стану дитини, яка може частково допомогти фахівцю у повсякденному житті виконувати частину його роботи, а також провести первинний аналіз емоційного стану дитини онлайн, без участі спеціалістів.

Об'єктом дослідження є ознаки та закономірності дитячих малюнків, за якими можна зробити висновок про емоційне здоров'я дитини.

Предметом дослідження є нейромережеві моделі та алгоритми, які можуть бути використані для виявлення характерних ознак на дитячих малюнках, що дасть змогу зробити висновки про емоційний та психологічний стан дитини.

Дослідження базується на аналізі результатів сучасних теоретичних і прикладних розробок вітчизняних і зарубіжних учених в ІТ галузі. Для вирішення поставлених завдань використано: методи побудови нейронних мереж, аналіз існуючих нейронних мереж та підходів до задач класифікації, методи об'єктно-орієнтованого програмування, методи побудови програмних застосунків з командним інтерфейсом.

Метою даної роботи є розробка нейромережевої моделі аналізу дитячих малюнків для діагностики емоційного стану дитини, яка може бути корисною як у вирішенні повсякденних питань фахівців, так і у персональному використанні для проведення первинної діагностики емоційного стану дитини.

Вимогами до системи є:

- навчатися на підібраному датасеті;
- приймати на вхід дитяче зображення;
- на виході віддавати результат аналізу щодо емоційного стану дитини;
- мати високу ймовірність вірного результату.

У першому розділі розглянуто аналіз предметної області і були поставлені задачі дослідження. Було розглянуто та досліджено яким саме

чином фахівці аналізують дитячі малюнки, на що вони звертають увагу, що є характерними ознаками того чи іншого емоційного стану. Також проаналізовано використання нейронних мереж для вирішення подібних задач. Розглянуто взагалі питання нейронних мереж, принципу їх роботи. Нейронна мережа — це серія алгоритмів, які намагаються розпізнати базові зв'язки в наборі даних за допомогою процесу, який імітує роботу людського мозку. У цьому сенсі нейронні мережі відносяться до систем нейронів, органічних або штучних за своєю природою. Розглянуто структуру нейронної мережі, яка складається із трьох компонентів: вхідного шару, прихованих (обчислювальних) шарів та вихідного шару. Проаналізовано базовий принцип навчання нейронної мережі, адже в ході роботи також потрібно буде навчати нейромережеву модель на підготовленому датасеті. Модель дізнається, які зв'язки між нейронами є важливими для успішного прогнозування під час навчання. На кожному етапі тренування мережа використовує математичну функцію, щоб визначити, наскільки точним був останній прогноз порівняно з очікуваним. Ця функція генерує серію значень помилок, які можуть використовуватися системою для розрахунку того, як модель повинна оновлювати значення ваги, прикріплених до кожного посилення, з кінцевою метою підвищення точності прогнозів мережі. Протягом багатьох тренувальних циклів і за допомогою періодичного ручного налаштування параметрів мережа продовжуватиме генерувати все більш точні прогнози, доки не досягне максимального значення. Навчити нейронну мережу можна різними способами: з вчителем, без вчителя, з підкріпленням. При навчанні з учителем нейронна мережа навчається на розміченому наборі даних і передбачає відповіді, які використовуються для оцінки точності алгоритму на навчальних даних. Під час навчання без вчителя модель використовує нерозмічені дані, у тому числі алгоритм самостійно намагається витягти ознаки і залежності. Навчання з частковим залученням вчителя є чимось середнім. Воно використовує невелику кількість розмічених даних та великий набір нерозмічених. А навчання із

підкріпленням тренує алгоритм за допомогою системи заохочень. Агент отримує зворотний зв'язок у вигляді винагород за правильні дії. Подібним чином дресирують тварин. Також було розглянуте питання переваг і недоліків нейронних мереж. Нейронні мережі, які можуть працювати безперервно є більш ефективними, ніж люди або простіші аналітичні моделі. Нейронні мережі також можна запрограмувати на навчання на попередніх виходах для визначення майбутніх результатів на основі подібності до попередніх вхідних даних. Із недоліків варто відзначити кількість даних, які їм потрібні для навчання. Іноді для досягнення високої точності роботи алгоритму датасети досягають дійсно величезних розмірів: нещодавно Facebook оголосив, що використав один мільярд зображень для досягнення рекордної продуктивності системи розпізнавання зображень. Проаналізовано існуючі типи нейронних мереж. Для різних даних і програм використовуються різні типи нейронних мереж. Різні архітектури спеціально розроблені для роботи з певними типами даних або доменом. Проведено аналіз від найпростіших до більш складних. Найголовніше, що було зроблено в ході першого розділу — це була обрана необхідна нейронна мережа для вирішення задачі роботи. Найкращі результати в області розпізнавання та класифікації об'єктів показала Convolutional Neural Network або згортова нейронна мережа. Успіх обумовлений можливістю обліку двовимірної топології зображення, на відміну багат шарового персептрона. Згортові нейронні мережі забезпечують часткову стійкість до змін зсувів, ракурсу, масштабу, поворотів та інших спотворень, що дійсно підходить для вирішення нашої задачі класифікації об'єктів саме на дитячих малюнках.

У другому розділі було побудовано нейромережеву модель аналізу дитячих малюнків. Розглянуто і математично описано архітектуру згорткової нейронної мережі. Визначено вхідні та вихідні дані з мережі. Вхідними даними є кольоровими зображення типу JPEG, розміру 224x224 пікселів. Кожне зображення розбивається на 3 канали: червоний, синій, зелений. Таким чином виходить 3 зображення розміру 224x224 пікселів. Вихідний

шар пов'язаний із усіма нейронами попереднього шару. Кількість нейронів відповідає кількості класів, що розпізнаються, тобто 3: «Щастя», «Тривога і депресія» та «Гнів і насильство». Також у цьому розділі була обрана функція активації. У роботі як функція активації у прихованих та вихідному шарах застосовується гіперболічний тангенс, у згорткових шарах застосовується ReLU. Розглянуто алгоритм машинного навчання, попередню обробку даних, пряме та зворотнє поширення, архітектуру ResNet. Також було обрано алгоритм навчання нейронної моделі.

У третьому розділі йдеться про експериментальну реалізацію описаного у другому розділі. Спочатку було зібрано тестові дані. Так як задовільних під задачу тестових наборів даних знайдено не було, було прийнято рішення сформувати власні. Таким чином було сформовано датасет, який був використаний для навчання моделі. Він містить три категорії із зображеннями, кожне з яких належить до однієї з трьох категорій: «Щастя», «Тривога і депресія» або «Гнів і насильство». За допомогою мови Python та різних її бібліотек була створена програмна реалізація нейронної мережі, її навчання тощо. У останньому підрозділі можна побачити результати роботи мережі та переконатися, що модель досить чітко розподіляє дитячі малюнки на потрібні класи.

Таким чином в ході роботи була побудована нейромережева модель аналізу дитячих малюнків для діагностики емоційного стану дитини. Дана система представлена як консольний додаток. Дана система показала, що вона може виконувати поставлені завдання.

МАШИННЕ НАВЧАННЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ,
НЕЙРОННІ МЕРЕЖІ, ШТУЧНИЙ ІНТЕЛЕКТ, ЕМОЦІЙНИЙ СТАН,
ДАТАСЕТ, ЗАДАЧА КЛАСИФІКАЦІЇ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	155
ВСТУП	166
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	188
1.1 Аналіз підходів до діагностики емоційного стану дитини	Ошибка! Закладка не определена.
1.2 Аналіз використання нейронних мереж для аналізу дитячих малюнків.....	Ошибка! Закладка не определена. 9
1.3 Поняття нейромережевої моделі ...	Ошибка! Закладка не определена. 1
1.4 Структура нейронної мережі	Ошибка! Закладка не определена. 3
1.5 Робота нейронної мережі.....	Ошибка! Закладка не определена. 5
1.6 Навчання нейронної мережі.....	Ошибка! Закладка не определена. 6
1.7 Набір даних для навчання нейронної мережі	Ошибка! Закладка не определена.
1.7.1 Навчання нейронної мережі з учителем	Ошибка! Закладка не определена. 8
1.7.2 Навчання нейронної мережі без учителя.....	30
1.7.3 Навчання нейронної мережі із частковим залученням вчителя.....	32
1.7.4 Навчання нейронної мережі з підкріпленням	Ошибка! Закладка не определена.
1.8 Переваги та недоліки нейронних мереж	Ошибка! Закладка не определена. 5
1.8.1 Переваги нейронних мереж	Ошибка! Закладка не определена.
1.8.2 Недоліки нейронних мереж	Ошибка! Закладка не определена. 5
1.9 Типи нейронних мереж.....	Ошибка! Закладка не определена. 8
1.10 Вибір типу моделі нейронної мережі для аналізу дитячих малюнків.....	Ошибка! Закладка не определена. 4
1.11 Створення архітектури системи ..	Ошибка! Закладка не определена. 5
2 СТВОРЕННЯ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ АНАЛІЗУ ДИТЯЧИХ МАЛЮНКІВ	18
2.1 Пошук умов для визначення емоційного стану за малюнком	Ошибка! Закладка не определена.

2.2	Визначення емоційного стану за малюнком	Ошибка! Закладка не определена.
2.3	Побудова нейромережевої моделі аналізу дитячих малюнків	50
2.3.1	Топологія згорткової нейромережі	Ошибка! Закладка не определена.
2.3.2	Вхідний шар	Ошибка! Закладка не определена. 3
2.3.3	Згортковий шар	Ошибка! Закладка не определена.
2.3.4	Повнозв'язний шар	Ошибка! Закладка не определена. 7
2.3.5	Вихідний шар.....	58
2.3.6	Вибір функції активації	Ошибка! Закладка не определена. 9
2.4	Алгоритм машинного навчання.....	62
2.4.1	Попередня обробка даних	Ошибка! Закладка не определена.
2.4.2	Пряме та зворотнє поширення.....	63
2.4.3	Архітектура ResNet.....	Ошибка! Закладка не определена.
2.4.4	Навчання згорткової нейронної мережі.....	66
3	ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ.....	478
3.1	Вибір тестових даних.....	Ошибка! Закладка не определена. 8
3.2	Створення і тренування моделі	Ошибка! Закладка не определена. 3
3.3	Класифікація зображень навченою нейронною мережею	Ошибка! Закладка не определена.
3.3	Результати роботи програми.....	Ошибка! Закладка не определена. 2
	ВИСНОВКИ.....	90
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	922
	ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ РОБОТИ	954

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ЕС	—	Експертна система
CNN	—	Згорткова нейронна мережа
НМ	—	Нейронна мережа
ШНМ	—	Штучні нейронні мережі
ШІ	—	Штучний інтелект
БЗ	—	База знань
БД	—	База даних

ВСТУП

У наш час величезна кількість людей, а особливо дітей, постійно перебувають під психологічним тиском в результаті впливу однолітків, стресу на навчанні або роботі та впливу інформаційних систем. Іноді буває дуже важливо виявити переживання дитини, поки вони не переросли в панічні страхи, комплекси та агресивну, неконтрольовану поведінку дитини в цілому.

Своєчасне виявлення проблем та надання допомоги можуть стати вирішальним моментом у формуванні дитини як повноцінної особистості. Але не завжди легко знайти підхід до дитини і визначити проблему. Дуже часто психологи не можуть поставити точний діагноз і допомогти дитині у зв'язку з недовірою дітей до лікаря.

Психологічні тести на основі малюнків на сьогодні це один з найпопулярніших інструментів для аналізу особистості. Основним принципом їх роботи є те, що пацієнт проектує на папір через малюнок свій психологічний стан, відображає власний характер, розкриває почуття та пережиті емоції, бажання. Якщо порівнювати цей спосіб з іншими варіантами дослідження особистості, то тести на основі малюнків мають ряд переваг, як то відсутність почуття страху у пацієнта в процесі опитування, що дозволяє отримати більш точну і об'єктивну оцінку його персональних особливостей, психологічного та емоційного стану.

З іншого боку, даний спосіб аналізу особистості має і свої мінуси, один з яких це те, що правильно розшифрувати і визначити наявність або відсутність проблем у випробуваного може тільки фахівець з профільною освітою.

У суспільстві, особливо для батьків, питання емоційного здоров'я дитини безумовно є пріоритетним питанням, а отже і тема аналізу емоційного стану дитини за її малюнками є актуальною.

Сьогодні все менше подиву викликає інтенсивна інтеграція нейронних мереж у повсякдення. Нейронна мережа — це метод штучного інтелекту, який вчить комп'ютери обробляти дані таким самим способом, як і людський мозок. Це тип процесу машинного навчання, що називається глибоким навчанням, який використовує взаємопов'язані вузли або нейрони в шаруватій структурі, що нагадує людський мозок. Він створює адаптивну систему, за допомогою якої комп'ютери навчаються на своїх помилках та постійно вдосконалюються. Таким чином, штучні нейронні мережі намагаються вирішувати складні завдання, такі як резюмування документів або розпізнавання осіб з більш високою точністю.

Нейронні мережі допомагають комп'ютерам приймати розумні рішення з обмеженою участю людини. Вони можуть вивчати та моделювати відносини між нелінійними та складними вхідними та вихідними даними. Нейронні мережі можуть розуміти неструктуровані дані та робити загальні спостереження без спеціального навчання. Наприклад, вони можуть розпізнати, що питання «Чи можу я замовити таксі?» та «Довезете до пункту А?» означають одне й те саме.

Таким чином, метою цієї роботи є побудова нейромережевої моделі аналізу дитячих малюнків для діагностики емоційного стану дитини, яка зможе отримати на вхід дитячий малюнок, проаналізувати його, зробити свої висновки і головне відповісти на питання «Чи здоровий емоційний стан дитини, яка малювала рисунок?». Аналізуючи зображення, що буде подаватися на вхід, виокремлюючи певні характерні ознаки того чи іншого малюнка, побудована модель допоможе скласти певну картину про емоційний стан дитини. Така модель допоможе провести первинний аналіз стану дитини без втручання лікаря, або стане у пригоді безпосередньо фахівцю.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз підходів до діагностики емоційного стану дитини

Велику увагу вчителів і вихователів привертала дитячі малюнки. Вираз дитини є частина його природи. Коли діти малюють, вони ретельно обирають матеріали, олівці, кольори, візерунки, розмір і положення того, що вони хочуть намалювати. Дитячі малюнки унікальні і можуть дати нам точну інформацію про молодого художника. Вивчення дитячих малюнків бере свій початок з кінця 19 століття (Thomas & Silk, 1990). З тих пір дослідження використовувалося в основному з естетичних, освітніх і клінічних міркувань. В основному є три різні види дослідження емоційно-експресивної сторони дитячого малюнка. По-перше, малюнки аналізуються як прояв особистісних рис, інтерпретованих в основному Фрейдом, в теоретичних рамках психоаналітичної теорії та її похідних. Другий, визначений переважно в роботі Коппіца (1968), який намагався розробити та науково підтвердити класифікацію «емоційних індикаторів», які можна знайти в дитячому малюнку. Третій вид був стурбований тим, як звичайні діти зображують особисто важливі чи емоційно важливі теми, а не оцінкою особистості чи клінічного діагнозу [1].

Розглянемо більш детально інтерпретації дитячих малюнків:

- малюнок дуже великої людини. Зазвичай величезний малюнок вказує на агресію та надмірну активність.
- малюнок дуже маленької людини. Зазвичай кажуть про маленькі малюнки людської фігури, що вони демонструють почуття некомпетентності, сорому, страху та депресії;
- голова. Якщо дитина збільшує голову, це свідчить про те, що її ego роздувається. Звичайні діти малюють голову відповідно до тіла;

- тонкі лінії. Вкладають на низький рівень фізичної та психічної енергії, сором і депресію;

- фігури. Під час малювання фігур, розмір і відносний розмір намальованих фігур вважаються значущими, при цьому більш важливі чи домінуючі фігури малюються більшими. Імпульсивні діти малюють великі фігури, відсутність шиї, асиметрію кінцівок; тривожні діти зображують хмари, дощ, літаючих птахів; сердиті діти малюють великі руки і зуби, довгі руки, схрещені очі.

Когнітивний розвиток і навички малювання є важливими факторами, які слід враховувати під час аналізу дитячих малюнків. Інформація, яку діти мають про екологічну проблему, буде вирішальною для її уявлення. Таким чином, дитячі малюнки можуть надати цінну інформацію про розвиток у дітей сприйняття навколишнього середовища. Стратегії, які використовують діти для малювання, представляють інтерес не тільки тому, що вони дозволяють нам зробити більш задовільний аналіз дитячої творчості, але також тому, що вони відкривають можливість прогресу в розумінні розвитку навичок планування та організації в цілому.

1.2 Аналіз використання нейронних мереж для аналізу дитячих малюнків

Штучні нейронні мережі (ШНМ) на сьогоднішній день пройшли три фази, перша з яких почалася в кібернетиці в 1940-1960-х роках. Однак вони були не в змозі впоратися з «гетерогенними» проблемами, і комп'ютери не мали достатньої обчислювальної потужності для запуску нейронних мереж протягом тривалого часу. Другий етап розпочався наприкінці 1980-х років з алгоритмом зворотного поширення Румельхарта, який подолав проблему «гетероскедастичності» та зменшив кількість складних обчислень, необхідних для двошарових нейронних мереж. У 2006 році Хінтон запропонував модель нейронної мережі під назвою Deep Belief Network

(DBN) для досягнення зменшення розмірності даних. Його основні ідеї включають: структури нейронної мережі з більшою кількістю прихованих шарів, що мають унікальні можливості вивчення функцій і можуть краще захоплювати основні характеристики зображень; труднощі навчання глибоких нейронних мереж можна подолати за допомогою ініціалізації «рівень за шаром».

Глибоке навчання — це загальний термін для класу методів, які навчають моделі з глибокою структурою. Основні моделі глибокого навчання включають мережі глибоких переконань (DBN), рекурентні нейронні мережі (RNN) і згорткові нейронні мережі (CNN). У 1998 році Янн запропонував заснований на градієнтному навчанні алгоритм CNN і застосував його для розпізнавання рукописних цифрових символів. CNN спочатку був натхненний нейронаукою та обробкою візуальної інформації простими та складними клітинами зорового нерву. Вони зазвичай використовують операції згортки, щоб імітувати обробку крайової інформації в різних напрямках простими клітинками, і операції об'єднання, щоб імітувати кумулятивну обробку подібних простих клітинок складними клітинками. CNN підтримує використання значень пікселів зображення безпосередньо як вхідних даних для неявного отримання інформації про абстрактні характеристики зображення без попередньої обробки зображення та явного виділення характеристик зображення, уникаючи складного виділення ознак і процесу вибору вручну. CNN є дуже надійним для перетворень (наприклад, трансляції, масштабування) і повороту зображень, а також чутливих проблем (наприклад, освітлення та оклюзія). Класифікація мальованих зображень за допомогою вектора Фішера, запропонованого Шнайдером та ін. досягла кращих характеристик розпізнавання, і точність розпізнавання може бути близькою до людської. У 2015 році Ю та інші запропонували Sketch-a-Net, який спеціально розроблений для розпізнавання мальованих зображень. Він застосовує злиття багатомасштабних мереж за допомогою байєсівського злиття, яке може ефективно вирішувати проблеми вилучення та

розрідженості намальованих вручну ескізів.

Квальї вдалося розпізнати емоції зображення на основі кольору та текстури. Бадрулхішам об'єднав візуальні функції, такі як колір, текстура та форма, і використав машину опорних векторів (SVM), щоб досягти відображення всебічного простору функцій зображення в простір емоцій. Фаррохі зазначив, що роль візуальних особливостей, таких як форма, колір і текстура зображень, у сприйнятті призведе до різних психологічних реакцій. Рідл припустив, що різні комбінації кольорів створюють різні емоції зображення. Переглядаючи літературу, пов'язану з класифікацією емоцій зображення, вказав, що існуючий аналіз емоцій зображення здебільшого базується на вивченні візуальних характеристик низького рівня, таких як колір, форма та текстура зображень. Тому, виходячи з принципів генерації та встановлених досліджень, на вивчення емоцій зображення зазвичай впливають візуальні характеристики низького рівня, такі як колір, текстура та форма [2].

1.3 Поняття нейромережевої моделі

Нейронна мережа — це серія алгоритмів, які намагаються розпізнати базові зв'язки в наборі даних за допомогою процесу, який імітує роботу людського мозку. У цьому сенсі нейронні мережі відносяться до систем нейронів, органічних або штучних за своєю природою.

Нейронні мережі можуть адаптуватися до змін вхідних даних; таким чином мережа генерує найкращий можливий результат без необхідності переробки критеріїв виводу. Концепція нейронних мереж, яка сягає своїм корінням у штучний інтелект, стрімко набирає популярність у розробці торгових систем.

Нейронні мережі є простими моделями того, як працює нервова система. Основними одиницями є нейрони, які зазвичай організовані в шари, як показано на рисунку 1.1.

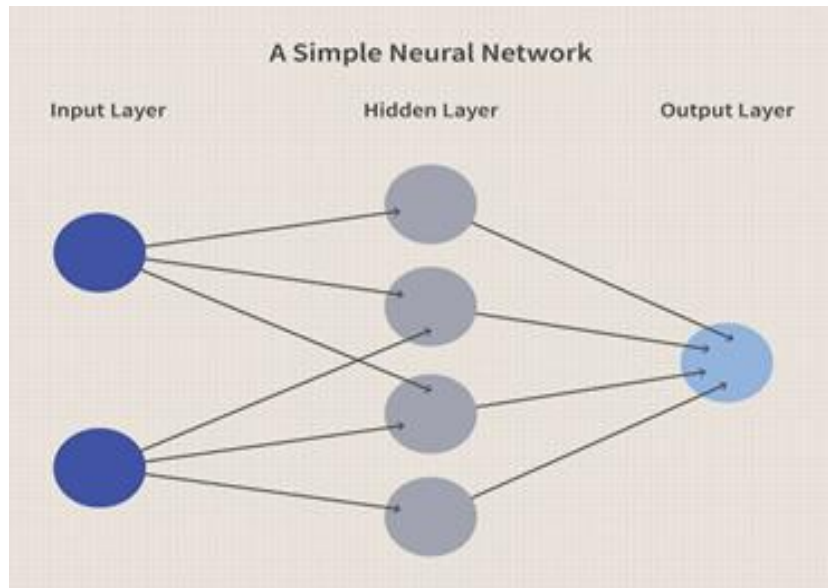


Рисунок 1.1 – Структура нейронної мережі

Нейронна мережа працює шляхом імітації великої кількості взаємопов'язаних блоків обробки, які нагадують абстрактні версії нейронів.

Блоки обробки розташовані пошарово. Зазвичай у нейронній мережі є три частини: вхідний рівень, з одиницями, що представляють поля введення; один або кілька прихованих шарів; і вихідний рівень з одиницею або одиницями, що представляють цільове поле(я). Агрегати з'єднані з різною силою з'єднання (або вагою). Вхідні дані подаються на перший рівень, а значення передаються від кожного нейрона до кожного нейрона наступного рівня. Зрештою, результат доставляється з вихідного рівня.

Мережа навчається, досліджуючи окремі записи, генеруючи прогноз для кожного запису та вносячи коригування вагових коефіцієнтів щоразу, коли робиться неправильний прогноз. Цей процес повторюється багато разів, і мережа продовжує вдосконалювати свої прогнози, доки не буде виконано один або кілька критеріїв зупинки.

Спочатку всі ваги є випадковими, і відповіді, які надходять із мережі, ймовірно, безглузді. Мережа вчиться через навчання. Приклади, для яких результат відомий, неодноразово подаються в мережу, а отримані відповіді

порівнюються з відомими результатами. Інформація з цього порівняння передається назад через мережу, поступово змінюючи ваги. Під час навчання мережа стає дедалі точнішою у відтворенні відомих результатів. Після навчання мережу можна застосовувати до майбутніх випадків, коли результат невідомий [3-4].

1.4 Структура нейронної мережі

Як вже було зазначено в попередньому розділі, нейронна мережа зазвичай організована шарами. Рівні складаються з багатьох взаємопов'язаних «вузлів», які містять «функцію активації». Нейронна мережа може містити наступні 3 рівні:

- вхідний шар. Призначення вхідного рівня — отримати на вхід значення пояснювальних атрибутів для кожного спостереження. Зазвичай кількість вхідних вузлів у вхідному шарі дорівнює кількості пояснювальних змінних. «Вхідний рівень» надає шаблони в мережу, які передаються в один або більше «приховані рівні». Вузли вхідного рівня пасивні, тобто вони не змінюють дані. Вони отримують єдине значення на своєму вході та дублюють значення на своїх багатьох виходах. З вхідного рівня він дублює кожне значення та надсилає до всіх прихованих вузлів;

- прихований шар. Вони застосовують задані перетворення до вхідних значень усередині мережі. У цьому випадку вхідні дуги, що йдуть від інших прихованих вузлів або від вхідних вузлів, з'єднаних з кожним вузлом. Він з'єднується з вихідними дугами до вихідних вузлів або до інших прихованих вузлів. У прихованому шарі фактична обробка здійснюється за допомогою системи зважених «з'єднань». Може бути один або декілька прихованих шарів. Значення, що надходять у прихований вузол, помножуються на ваги, набір заздалегідь визначених чисел, що зберігаються в програмі. Потім зважені вхідні дані додаються, щоб отримати одне число;

- вихідний шар. Потім приховані шари зв'язуються з «вихідним

шаром». Вихідний рівень отримує з'єднання з прихованих шарів або з вхідного рівня. Він повертає вихідне значення, яке відповідає передбаченню змінної відповіді. У задачах класифікації зазвичай є лише один вихідний вузол. Активні вузли вихідного рівня об'єднують і змінюють дані для отримання вихідних значень.

Здатність нейронної мережі забезпечувати корисну маніпуляцію даними полягає в правильному виборі вагових коефіцієнтів. Це відрізняється від звичайної обробки інформації.

Підсумовуючи, подібна природньому аналогу штучна нейромережа складається з нейронів і синапсів. Нейрон — це одиниця, яка отримує інформацію і виконує певні обчислення. Він є простішою структурною одиницею будь-якої нейромережі. Як правило, нейрони впорядковуються в шари, які зрештою формують мережу. Усі нейрони працюють приблизно однаково. Однак існують деякі часткові випадки нейронів, що виконують специфічні функції. Основні типи нейронів: вхідний (input) — шар нейронів, що отримує інформацію; прихований (hidden) — кілька шарів, що обробляють інформацію; вихідний (output) — шар нейронів, що представляє результати обчислень (рис. 1.2).

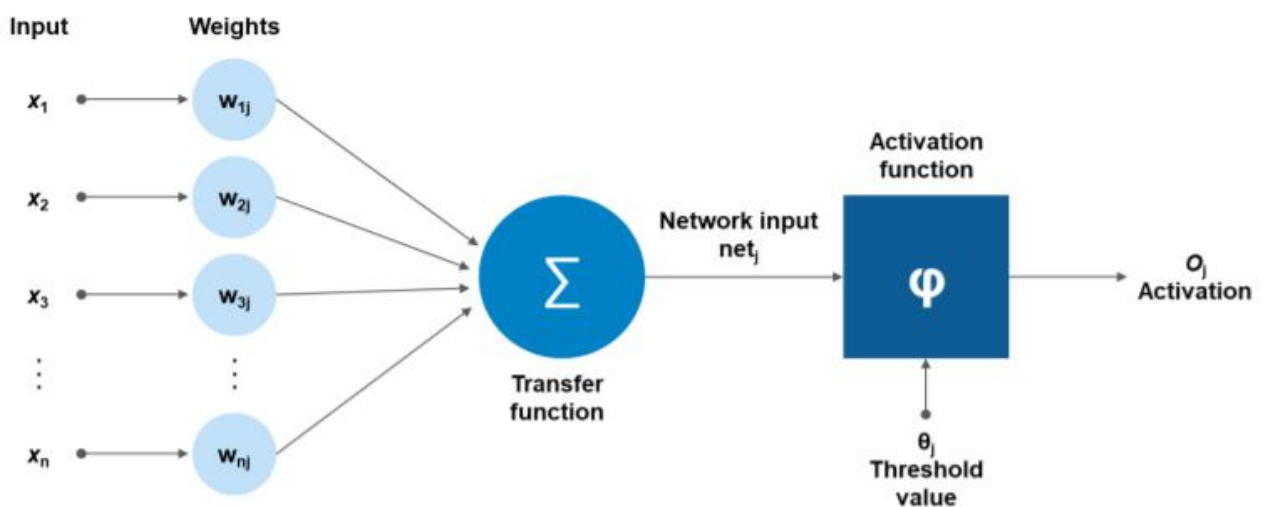


Рисунок 1.2 – Модель штучного нейрона

Синапс — це зв'язок, що поєднує вихід одного нейрона з входом іншого. Сигнал, що проходить через нього, може посилюватися або слабшати. Параметром синапсу є вага — коефіцієнт, завдяки якому інформація що передається з одного нейрона до іншого може змінюватися.

Важливу роль в архітектурі нейромережі грає активатор. Як і в мозку живого організму, він вирішує, які сигнали пропускати через нейрони, а які ні. Наприклад, ви взялися за гарячий чайник. Нервові закінчення на пальцях передадуть інформацію до нейронів головного мозку, де функція активації прийме рішення: відсмикнути руку від джерела тепла або продовжувати пропускати сигнали.

1.5 Робота нейронної мережі

На вхідний шар нейронів надходить деяка інформація, яка по синапсам переходить на наступний шар. При цьому кожний синапс має власну коефіцієнтну вагу, а будь-який наступний нейрон у новому шарі може мати кілька входів. Інформація передається далі, поки не дійде до кінцевого виходу.

Наприклад, алгоритм розпізнавання рукописного тексту повинен мати можливість впоратись з величезною різноманітністю способів представлення даних. Кожну цифру від 0 до 9 можна записати безліччю способів: розмір і точна форма кожного символу можуть відрізнятися залежно від того, хто пише і в яких обставинах.

Вхідному шару подаються значення, що становлять пікселі, які складають зображення рукописної цифри. Вихідний шар, у свою чергу, передбачає, який символ зображений на картинці.

Круги є нейронами, які організовані у зв'язані між собою вертикальні шари. Кольори посилянь також різняться: вони позначають важливість зв'язків між нейронами. Червоні зв'язки посилюють значення при переході

між шарами, що збільшує шанс активації нейрона, в який надходить значення. Активовані нейрони заштриховані червоним. У «Прихованому шарі» вони означають, що зображення рукописної фігури містить певну комбінацію пікселів, що нагадують горизонтальну лінію зверху рукописного числа 3 або 7 (рис. 1.3).

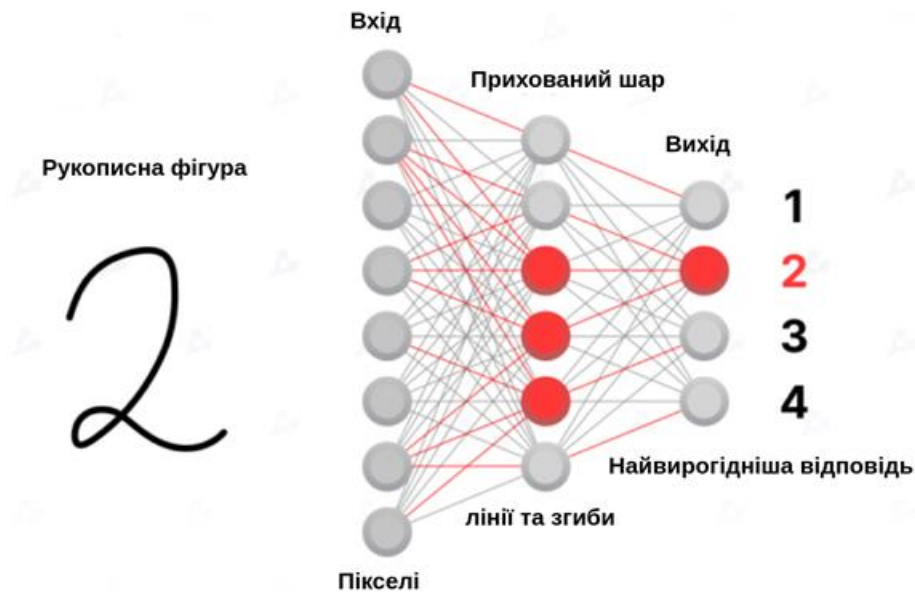


Рисунок 1.3 – Робота нейронної мережі

Таким чином, «Прихований шар» може виявити характерні лінії та криві, які зрештою об'єднуються в повну рукописну фігуру.

1.6 Навчання нейронної мережі

Модель дізнається, які зв'язки між нейронами є важливими для успішного прогнозування під час навчання. На кожному етапі тренування мережа використовує математичну функцію, щоб визначити, наскільки точним був останній прогноз порівняно з очікуванням.

Ця функція генерує серію значень помилок, які можуть використовуватися системою для розрахунку того, як модель повинна

оновлювати значення ваги, прикріплених до кожного посилання, з кінцевою метою підвищення точності прогнозів мережі.

Протягом багатьох тренувальних циклів і за допомогою періодичного ручного налаштування параметрів мережа продовжуватиме генерувати все більш точні прогнози, доки не досягне максимального значення. На цьому етапі, наприклад, коли рукописні цифри можна розпізнати з точністю понад 95%, можна сказати, що нейромережа навчена (рис.1.4).

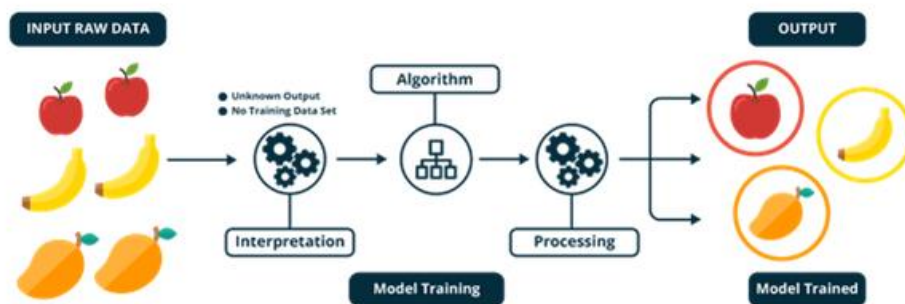


Рисунок 1.4 – Результат навчання нейронної мережі - кластеризація зображень

1.7 Набір даних для навчання нейронної мережі

Набір даних чи датасет — це колекція однотипних даних, необхідних для навчання нейромереж. Наприклад, щоб натренувати алгоритм розпізнавання людських облич, йому необхідно показати велику кількість фотографій інших людей. Чим більше даних – тим точніше алгоритм. Дані можуть бути будь-якого формату: таблиці, фото, відео, аудіо та інше.

Датасети бувають трьох видів:

- тренувальний (training dataset) — використовується при навчанні нейромережі;
- валідаційний (validation dataset) — незалежний набір даних, що використовується для фінальної оцінки точності алгоритму;

- тестовий (test dataset) — необхідний для перевірки точності навчання.

Навчити нейронну мережу можна різними способами: з вчителем, без вчителя, з підкріпленням. Але як вибрати оптимальний алгоритм і чим вони відрізняються?

При навчанні з учителем нейронна мережа навчається на розміченому наборі даних і передбачає відповіді, які використовуються для оцінки точності алгоритму на навчальних даних. Під час навчання без вчителя модель використовує нерозмічені дані, у тому числі алгоритм самостійно намагається визначити ознаки і залежності.

Навчання з частковим залученням вчителя є чимось середнім. Воно використовує невелику кількість розмічених даних та великий набір нерозмічених.

А навчання із підкріпленням тренує алгоритм за допомогою системи заохочень. Агент отримує зворотний зв'язок у вигляді винагород за правильні дії. Подібним чином дресирують тварин.

Для кожного способу навчання розглянемо приклади відповідних йому даних і завдань.

1.7.1 Навчання нейронної мережі з учителем

Навчання з учителем (supervised learning) передбачає наявність повного набору розмічених даних для тренування моделі на всіх етапах її побудови.

Наявність повністю розміченого датасета означає, що кожному прикладу в навчальному наборі співвідноситься відповідь, яку алгоритм повинен отримати. Таким чином, розмічений датасет із фотографій квітів навчить нейронну мережу, де зображені троянди, ромашки чи нарциси. Коли мережа отримає нове фото, вона порівняє його з прикладами навчального датасету, щоб передбачити відповідь.

В основному навчання з учителем застосовується для вирішення двох типів

завдань: класифікації та регресії.

В задачах класифікації алгоритм передбачає дискретні значення, які відповідають номерам класів, до яких належать об'єкти. У датасеті з фотографіями тварин кожне зображення матиме відповідну мітку — «кішка», «коала» або «черепашка». Якість алгоритму оцінюється тим, як точно він може правильно класифікувати нові фото з коалами і черепахами (рис. 1.5).

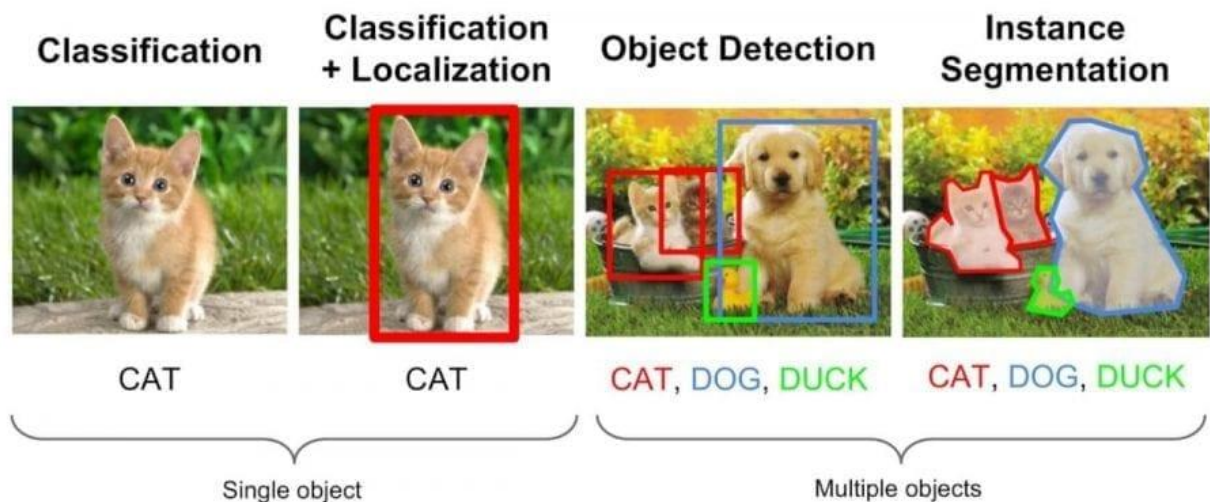


Рисунок 1.5 – Приклад навчання з учителем – класифікація (ліворуч), та подальше її використання для сегментації та розпізнавання об'єктів

Завдання регресії пов'язані із безперервними даними. Наприклад, лінійна регресія обчислює очікуване значення змінної y , враховуючи конкретні значення x . Більш утилітарні завдання машинного навчання задіюють велику кількість змінних: наприклад, нейронна мережа, що передбачає ціну квартири в Сан-Франциско на основі її площі, розташування та доступності громадського транспорту. Алгоритм виконує роботу експерта, який розраховує ціну квартири виходячи з тих самих даних. Таким чином, навчання з учителем найбільше підходить для завдань, коли є значний набір достовірних даних для навчання алгоритму.

1.7.2 Навчання нейронної мережі без учителя

Ідеально розмічені та чисті дані дістати нелегко. Тому іноді перед алгоритмом стоїть завдання знайти заздалегідь невідомі відповіді. Ось де потрібне навчання без вчителя. У навчанні без вчителя (unsupervised learning) модель має набір даних, і немає явних вказівок, що з ним робити. Нейронна мережа намагається самостійно знайти кореляції даних, витягуючи корисні ознаки та аналізуючи їх.

Розглянемо більш детально інтерпретації дитячих малюнків:

- кластеризація. Навіть без спеціальних знань експерта-орнітолога можна подивитися на колекцію фотографій та розділити їх на групи за видами котів, спираючись на колір, розмір або форму вух. Саме в цьому полягає кластеризація – найпоширеніша задача для навчання без учителя. Алгоритм підбирає схожі дані, знаходячи загальні ознаки і групує їх разом;

- виявлення аномалій. Банки можуть виявити шахрайські операції, виявляючи незвичайні дії у купівельній поведінці клієнтів. Наприклад, підозріло, якщо одна кредитна картка використовується в США та Європі в той самий день. Схожим чином, навчання без вчителя використовують для знаходження нетипових даних;

- асоціації. Виберете в онлайн-магазині підгузки, яблучне пюре та дитячий кухоль-непроливайку та сайт порекомендує вам додати нагрудник та радіоняню до замовлення. Це приклад асоціацій: деякі характеристики об'єкта корелюють з іншими ознаками. Розглядаючи декілька ключових ознак об'єкта, модель може передбачити інші, із якими існує зв'язок;

- автоенкодер. Автоенкодери приймають вхідні дані, кодують їх, а потім намагаються відтворити початкові дані з отриманого коду. Не так багато реальних ситуацій, коли використовують простий автоенкодер. Але варто додати шари та можливості розширяться: використовуючи зашумлені та вихідні версії зображень для навчання, автоенкодери можуть видаляти шум із відеоданих, зображень або медичних сканів, щоб підвищити якість даних.

У навчанні без вчителя складно обчислити точність алгоритму, оскільки в наборі даних відсутні «правильні відповіді» чи мітки. Розмічені дані часто ненадійні. У таких випадках, надаючи моделі свободу дій для пошуку залежностей, можна отримати непогані результати (рис.1.6).



Рисунок 1.6 – Кластеризація даних на основі загальних ознак

1.7.3 Навчання нейронної мережі із частковим залученням вчителя

Навчання з частковим залученням вчителя (semi-supervised learning) характеризується своєю назвою: навчальний датасет містить як розмічені, і нерозмічені дані. Цей метод особливо корисний, коли важко витягти з даних важливі ознаки або розмітити всі об'єкти – трудомістка задача.

Цей метод машинного навчання є поширеним для аналізу медичних зображень, таких як скани комп'ютерної томографії або МРТ. Досвідчений рентгенолог може розмітити невелику підмножину сканів, на яких виявлено пухлини та захворювання. Але вручну розмічати всі скани - надто трудомістке і дороге завдання. Тим не менш, нейронна мережа може

отримати інформацію з невеликої частки розмічених даних і поліпшити точність передбачень у порівнянні з моделлю, що навчається виключно на нерозмічених даних.

Популярний метод навчання, що потребує невеликий набір розмічених даних, полягає у використанні генеративно-змагальної мережі GAN. Дискримінатору показують вихідні зображення та дані, створені генератором. Він повинен визначити, які зображення є реальними, а які є підробленими (рис.1.7).

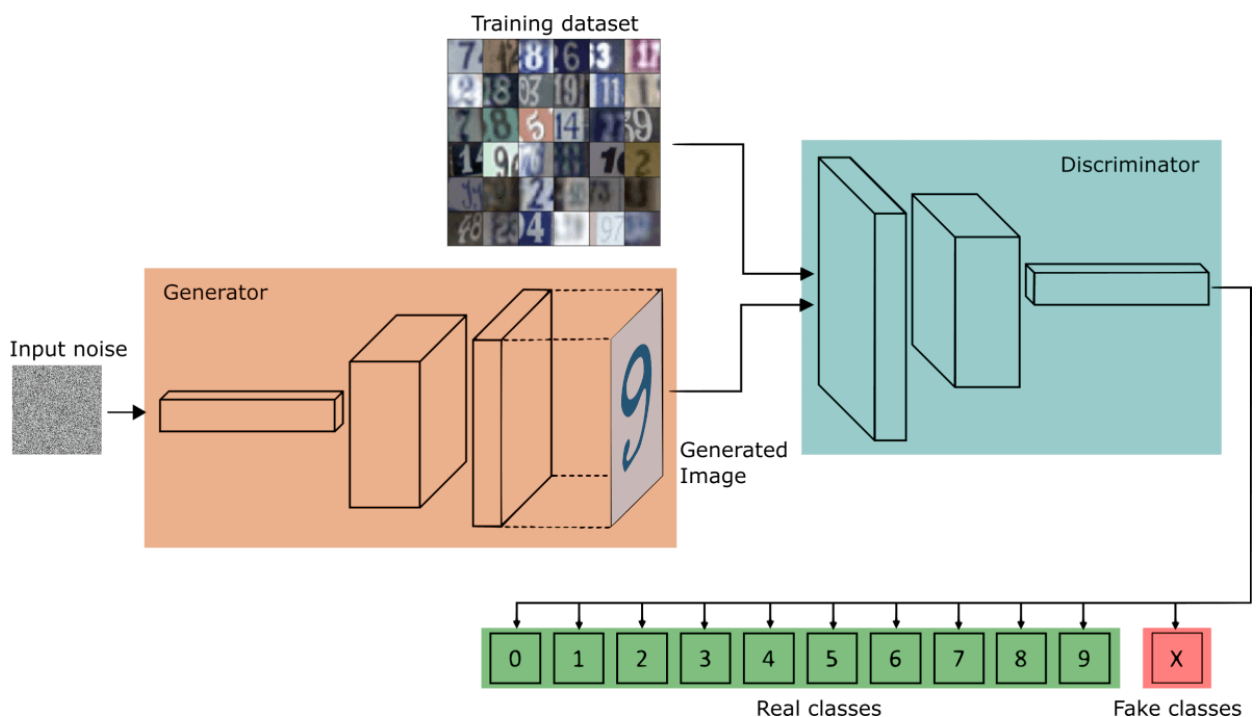


Рисунок 1.7 – Алгоритм роботи GAN

Уявіть собі змагання двох нейронних мереж, де кожна намагається перехитрити іншу. Це GAN. Одна з мереж, генератор, намагається створити нові об'єкти даних, які імітують навчальну вибірку. Інша мережа, дискримінатор, оцінює, чи ці згенеровані дані є реальними чи підробленими. Мережі взаємодіють і циклічно вдосконалюються, оскільки дискримінатор намагається краще відокремлювати підробки від оригіналів, а генератор намагається створювати переконливі підробки.

1.7.4 Навчання нейронної мережі з підкріпленням

Відеоігри ґрунтуються на системі стимулів. Завершіть рівень та отримайте нагороду. Переможіть всіх монстрів та заробите бонус. Потрапили у пастку – кінець гри, не влучайте. Ці стимули допомагають гравцям зрозуміти, як краще діяти у наступному раунді гри. Без зворотного зв'язку люди просто приймали б випадкові рішення і сподівалися перейти на наступний ігровий рівень.

Навчання з підкріпленням (reinforcement learning) діє за тим самим принципом. «Агент» проходить трасу, не виїжджаючи за її межі. Далі можна досягти підвищення швидкості проходження траси (рис.1.8).

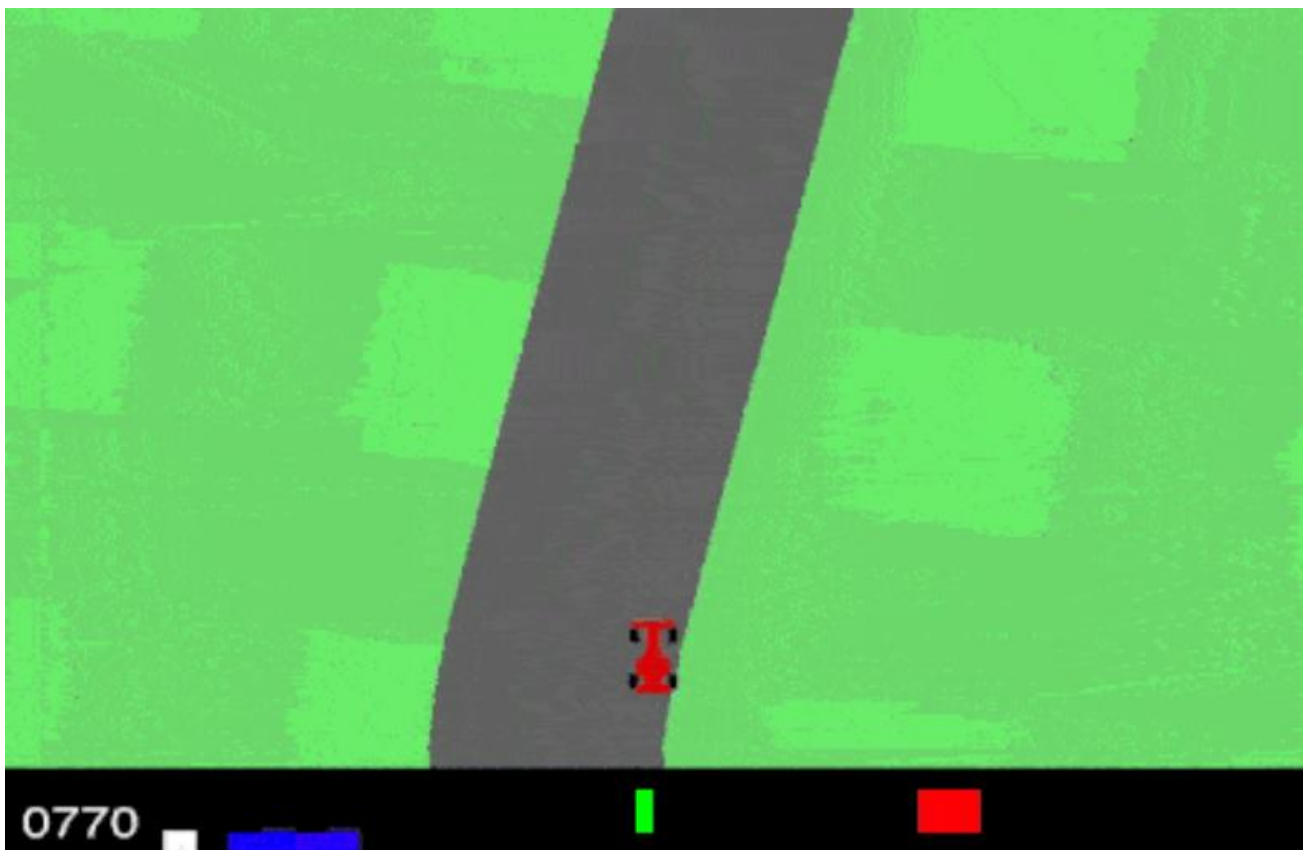


Рисунок 1.8 – Результат навчання з підкріпленням.

Агенти штучного інтелекту намагаються знайти оптимальний спосіб досягнення мети чи покращення продуктивності для конкретного середовища. Коли агент робить дії, що сприяють досягненню мети, він отримує нагороду. Глобальна мета – передбачати такі кроки, щоб зрештою заробити максимальну нагороду.

При ухваленні рішення агент вивчає зворотний зв'язок, нові тактики та рішення здатні призвести до більшого виграшу. Цей підхід використовує довгострокову стратегію — як і в шахах: наступний найкращий з усіх можливих хід може не допомогти виграти зрештою. Тож агент намагається максимізувати сумарну нагороду.

Це ітеративний процес. Чим більше рівнів із зворотного зв'язку, тим краще стає стратегія агента. Такий підхід є особливо корисним для навчання роботів, які керують автономними транспортними засобами або інвентарем на складі.

Так само, як і учні у школі, кожен алгоритм навчається по-різному. Але завдяки різноманітності доступних методів, питання в тому, щоб вибрати відповідний і навчити вашу нейронну мережу розбиратися в середовищі.

1.8 Переваги та недоліки нейронних мереж

1.8.1 Переваги нейронних мереж

Нейронні мережі, які можуть працювати безперервно є більш ефективними, ніж люди або простіші аналітичні моделі. Нейронні мережі також можна запрограмувати на навчання на попередніх виходах для визначення майбутніх результатів на основі подібності до попередніх вхідних даних.

Нейронні мережі, які використовують хмарні онлайн-сервіси, також мають переваги зменшення ризиків порівняно з системами, які покладаються на апаратне забезпечення локальних технологій. Крім того, нейронні мережі

часто можуть виконувати кілька завдань одночасно (або, принаймні, розподіляти завдання, які повинні виконуватися модульними мережами одночасно).

Нарешті, нейронні мережі постійно розширюються в нові програми. Хоча ранні теоретичні нейронні мережі були дуже обмежені в застосуванні в різних галузях, сьогодні нейронні мережі використовуються в медицині, науці, фінансах, сільському господарстві чи безпеці.

1.8.2 Недоліки нейронних мереж

Хоча нейтральні мережі можуть покладатися на онлайн-платформи, для створення нейронної мережі необхідний апаратний компонент. Це створює фізичний ризик для мережі, яка залежить від складних систем, вимог до налаштування та можливого фізичного обслуговування.

Хоча складність нейронних мереж є сильною стороною, це може означати, що на розробку певного алгоритму для конкретного завдання потрібні місяці (якщо не більше). Крім того, може бути важко помітити будь-які помилки або недоліки в процесі, особливо якщо результати є оцінками або теоретичними діапазонами.

Нейронні мережі також може бути важко перевірити. Деякі процеси нейронної мережі можуть виглядати як «чорна скринька», де вводяться вхідні дані, мережі виконують складні процеси та повідомляють про результати. Людям також може бути важко проаналізувати слабкі сторони в процесі обчислення або навчання мережі, якщо мережі бракує загальної прозорості щодо того, як модель вивчає попередні дії.

Одним із найбільших недоліків нейромереж є кількість даних, які їм потрібні для навчання. Іноді для досягнення високої точності роботи алгоритму датасети досягають дійсно величезних розмірів: нещодавно Facebook оголосив, що використав один мільярд зображень для досягнення рекордної продуктивності системи розпізнавання зображень.

Через розмір наборів даних та кількість циклів навчання часто потрібний доступ до потужного та коштовного комп'ютерного обладнання, забезпеченого високопродуктивними графічними процесорами. Незалежно від того, чи ви створюєте свою власну систему або орендуєте обладнання на хмарній платформі, це так чи інакше впливає на вартість навчання.

Іншим викликом для створення високоточних моделей є неточності наборів даних. Як уже зазначалося, люди можуть припускатися помилок при створенні датасетів, що тією чи іншою мірою може вплинути на підсумковий результат.

1.9 Типи нейронних мереж

Для різних даних і програм використовуються різні типи нейронних мереж. Різні архітектури нейронних мереж спеціально розроблені для роботи з певними типами даних або доменом. Почнемо з найпростіших і підемо до більш складних [5-7].

1.9.1 Перцептрон

Модель перцептрона, запропонована Мінським-Пейпертом, є однією з найпростіших і найстаріших моделей нейрона. Це найменша одиниця нейронної мережі, яка виконує певні обчислення для виявлення особливостей або бізнес-аналітики у вхідних даних. Він приймає зважені вхідні дані та застосовує функцію активації, щоб отримати результат як кінцевий результат. Перцептрон також відомий як TLU (порогова логічна одиниця) (рис. 1.9).

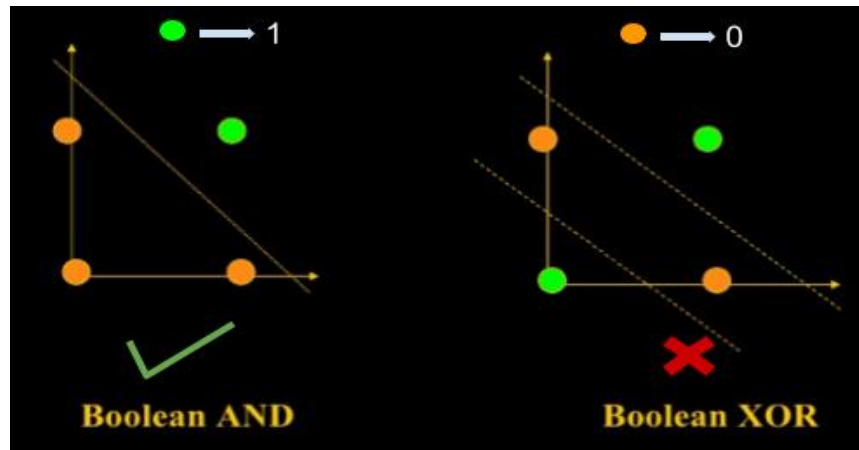


Рисунок 1.9 – Приклад роботи персептрона

Perceptron — це алгоритм навчання під наглядом, який класифікує дані за двома категоріями, отже, це двійковий класифікатор. Персептрони можуть реалізувати такі логічні оператори, як AND, OR або NAND. Персептрони можуть вивчати лише задачі з лінійним відокремленням. Для нелінійних задач, таких як логічна проблема XOR, це не працює.

1.9.2 Нейронні мережі прямого зв'язку

Найпростіша форма нейронних мереж, де вхідні дані рухаються лише в одному напрямку, проходячи через штучні нейронні вузли та виходячи через вихідні вузли (рис. 1.10).

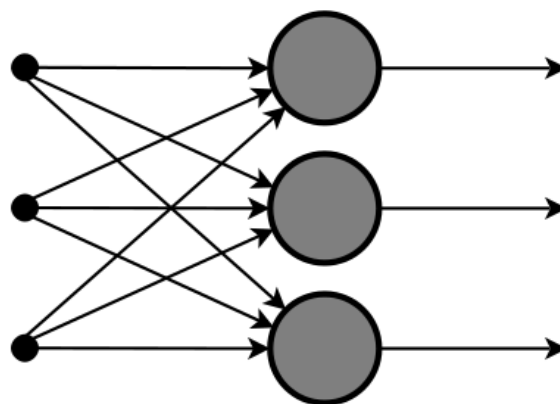


Рисунок 1.10 – Нейронна мережа прямого зв'язку

Там, де приховані шари можуть або не можуть бути присутніми, присутні вхідний і вихідний шари. Виходячи з цього, їх можна далі класифікувати як одношарові або багаторівневі нейронні мережі прямого зв'язку.

Кількість шарів залежить від складності функції. Вона має односпрямоване пряме поширення, але не має зворотного. Ваги тут статичні. Функція активації подається вхідними даними, які помножуються на ваги. Для цього використовується класифікуюча функція активації або функція покрокової активації. Наприклад: нейрон активується, якщо він перевищує порогове значення (зазвичай 0), і нейрон видає 1 як вихід. Нейрон не активується, якщо він нижче порогового значення (зазвичай 0), яке вважається 1. Ці мережі досить прості в обслуговуванні та обладнані для роботи з даними, що містять багато шуму. Найчастіше використовуються для таких задач: класифікація; розпізнавання мови, обличчя, образів.

1.9.3 Мережа радіальних базисних функцій

Радіальні базові мережі (RBN) передбачають необхідні цілі принципово іншим способом. Вони складаються з трьох шарів: вхідного, шару з нейронами RBF і вихідного. Фактичні класи на кожний приклад даних навчання зберігаються в нейронах RBF. Оскільки радіальна функція використовується як функція активації, RBN відрізняється від традиційного багатшарового персептрона (рис. 1.11).

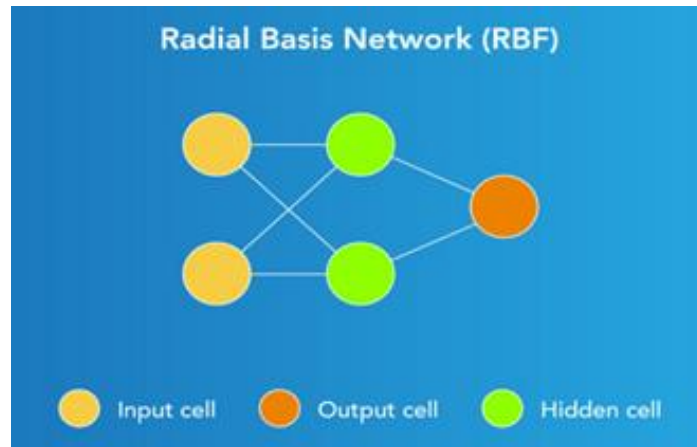


Рисунок 1.11 – Радіальна базисна нейронна мережа

Нейрони RBF перевіряють евклідову відстань значень ознак із фактичними класами, що зберігаються в нейронах, коли нові дані вводяться в нейронну мережу. Це можна порівняти з визначенням, до якого кластера належить конкретний екземпляр. Прогнозований клас присвоюється класу з найменшою відстанню. Найчастіше використовуються в системах відновлення запасу.

1.9.4 Багатошаровий перцептрон

Головним недоліком мереж Feed Forward була їх нездатність навчатися за допомогою зворотного поширення. Багатошарові перцептрони — це нейронні мережі, які містять кілька прихованих шарів і функції активації. Навчання відбувається під контролем, де ваги оновлюються за допомогою градієнтного спуску.

Багатошаровий перцептрон є двонаправленим, тобто вхідні дані передаються вперед, а вагові значення — у зворотному напрямку. Функції активації можна змінювати залежно від типу цілі. Softmax зазвичай використовується для багатокласової класифікації, Sigmoid для бінарної класифікації тощо. Їх також називають щільними мережами, оскільки всі нейрони шару з'єднані з усіма нейронами наступного шару.

1.9.5 Згорткові нейронні мережі

Коли справа доходить до класифікації зображень, найбільш використовуваними нейронними мережами є згорткові нейронні мережі (CNN). CNN містять кілька шарів згортки, які відповідають за вилучення важливих характеристик із зображення. Попередні рівні відповідають за деталі низького рівня, а пізніші рівні відповідають за функції більш високого рівня.

Операція Convolution використовує спеціальну матрицю, яку також називають фільтрами, для згортання вхідного зображення та створення карт. Ці фільтри ініціалізуються випадковим чином, а потім оновлюються за допомогою зворотного поширення. Одним із прикладів такого фільтра є Canny Edge Detector, який використовується для пошуку країв на будь-якому зображенні.

Після шару згортки є рівень об'єднання, який відповідає за агрегацію карт, отриманих із шару згортки. Це може бути максимальне об'єднання, мінімальне об'єднання тощо. Для регуляризації CNN також мають опцію для додавання шарів вилучення, які відкидають або роблять певні нейрони неактивними, щоб зменшити переобладнання та пришвидшити конвергенцію.

CNN використовують ReLU (Rectified Linear Unit) як функції активації в прихованих шарах. Як останній рівень, CNN мають повністю зв'язаний щільний шар, і функція активації здебільшого функціонує як Softmax для класифікації та переважно ReLU для регресії (рис. 1.12).

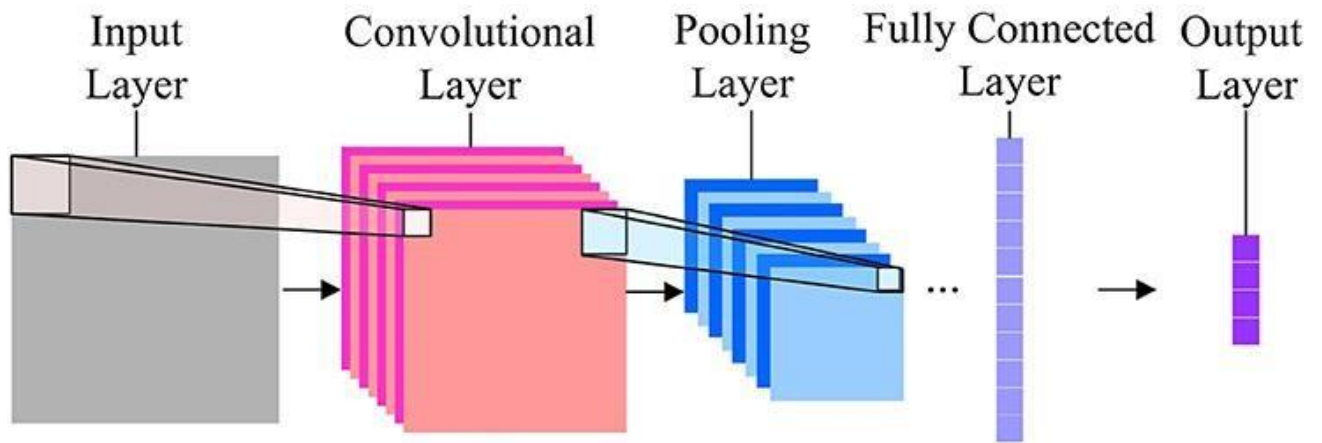


Рисунок 1.12 – Згорткова нейронна мережа

1.9.6 Рекурентні нейронні мережі

Більш складний тип нейронних мереж. Рекурентні нейронні мережі отримують вихідні дані вузла обробки та передають інформацію назад у мережу.

Це призводить до теоретичного «навчання» та вдосконалення мережі. Кожен вузол зберігає історичні процеси, і ці історичні процеси повторно використовуються в майбутньому під час обробки.

Це стає особливо критичним для мереж, у яких передбачення є неправильним; система намагатиметься дізнатися, чому стався правильний результат, і відповідно коригуватиметься. Цей тип нейронної мережі часто використовується в програмах перетворення тексту в мовлення (рис. 1.13).

Recurrent Neural Networks

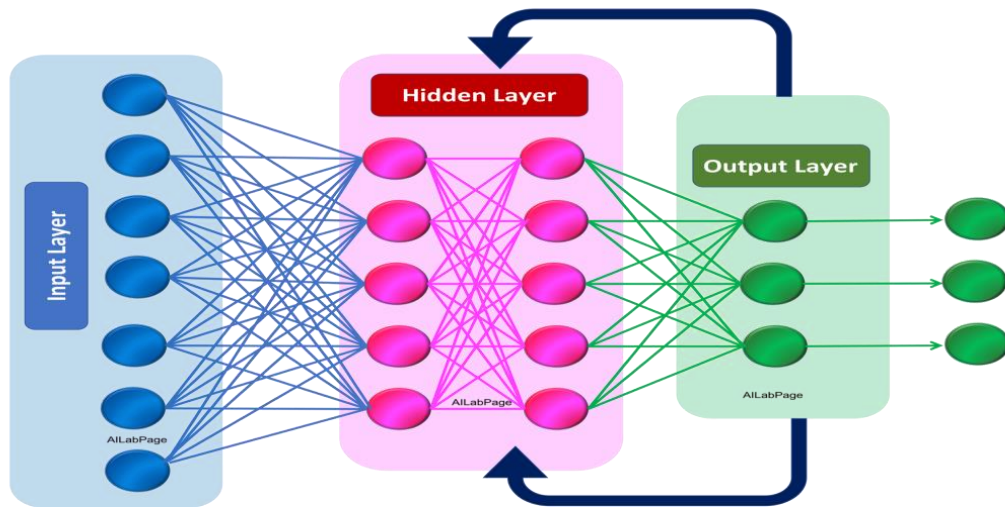


Рисунок 1.13 – Рекурентна нейронна мережа

1.9.7 Мережі LSTM (Довга короткочасна пам'ять)

Мережі LSTM є типом RNN, що використовують спеціальні одиниці на додаток до стандартних одиниць. Блоки LSTM містять «комірку пам'яті», яка може зберігати інформацію в пам'яті протягом тривалого часу. Набір воріт використовується для контролю, коли інформація надходить у пам'ять, виводиться, і коли вона забувається. Існує три типи воріт, а саме: вхідні, вихідні та забуті. Вхідний клапан визначає, скільки інформації з останнього зразка буде зберігатися в пам'яті; вихідний клапан регулює кількість даних, що передаються на наступний рівень, а клапани забуття контролюють швидкість розриву збереженої пам'яті. Ця архітектура дозволяє їм вивчати довгострокові залежності.

1.9.8 Моделі послідовності (Seq2seq)

Модель послідовності складається з двох рекурентних нейронних мереж. Тут існує кодер, який обробляє вхід, і декодер, який обробляє вихід.

Кодер і декодер працюють одночасно — або використовуючи той самий параметр, або різні. Ця модель, на відміну від реальної RNN, особливо доречна в тих випадках, коли довжина вхідних даних дорівнює довжині вихідних даних. Хоча вони мають ті ж переваги та обмеження, що й RNN, ці моделі зазвичай застосовуються в основному в чат-ботах, машинних перекладах і системах відповідей на запитання (рис. 1.14).

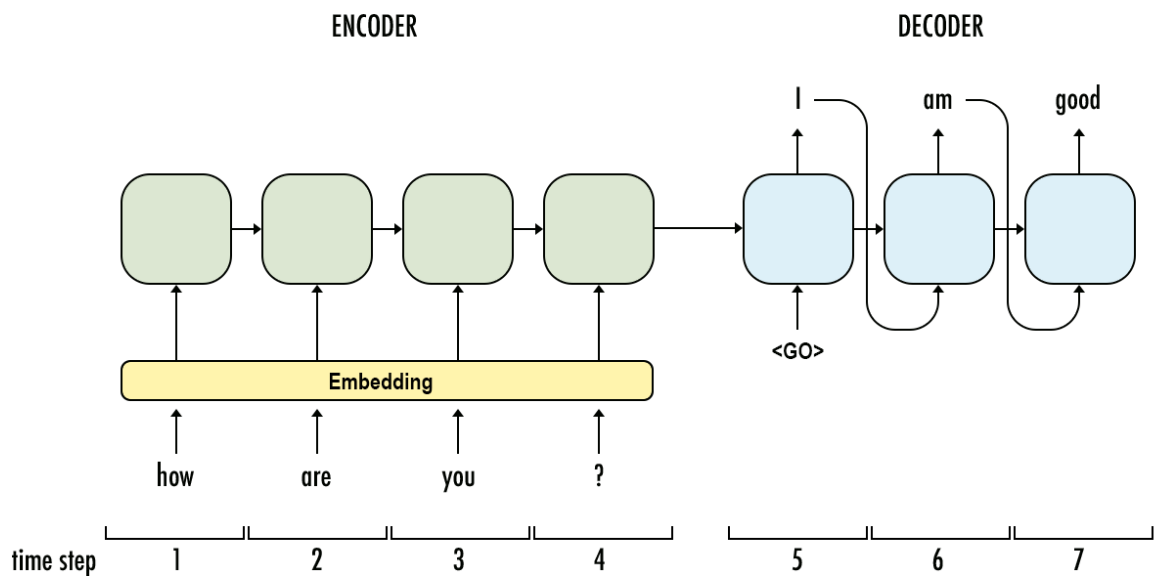


Рисунок 1.14 – Модель послідовності

1.9.9 Модульна нейронна мережа

Модульна нейронна мережа має ряд різних мереж, які функціонують незалежно та виконують підзавдання. Різні мережі насправді не взаємодіють і не сигналізують одна одній під час процесу обчислень. Вони самостійно працюють над досягненням результату.

У результаті великий і складний обчислювальний процес виконується значно швидше, розбиваючи його на незалежні компоненти. Швидкість обчислень збільшується, оскільки мережі не взаємодіють і навіть не з'єднані одна з одною. Із недоліків можна відзначити, що такі мережі мають проблеми

з розпізнаванням рухомих цілей (рис. 1.15).

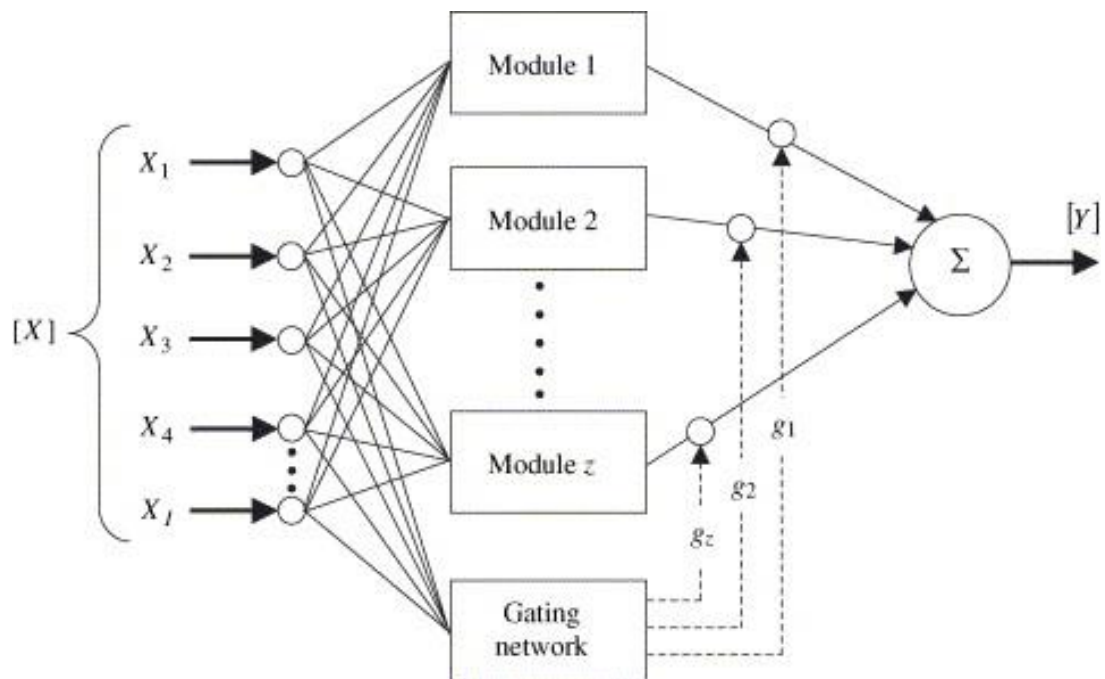


Рисунок 1.15 – Модульна нейронна мережа

1.10 Вибір типу моделі нейронної мережі для аналізу дитячих малюнків

Найкращі результати в області розпізнавання та класифікації об'єктів показала Convolutional Neural Network або згорткова нейронна мережа. Успіх обумовлений можливістю обліку двовимірної топології зображення, на відміну багат шарового перцептрона.

Згорткові нейронні мережі забезпечують часткову стійкість до змін зсувів, ракурсу, масштабу, поворотів та інших спотворень, що дійсно підходить для вирішення нашої задачі класифікації об'єктів саме на дитячих малюнках. Погодьтеся, що навіть дорослій людині не завжди зрозуміло що саме намалювала дитина, що вже казати про комп'ютери.

Згорткові нейронні мережі поєднують три архітектурні ідеї, для забезпечення інваріантності до зміни масштабу, повороту зсуву та

просторових спотворень:

- локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів);
- загальні синаптичні коефіцієнти (забезпечують детектування деяких рис в будь-якому місці зображення та зменшують загальну кількість вагових коефіцієнтів);
- ієрархічна організація із просторовими підвиборками;

На даний момент згортова нейронна мережа та її модифікації вважаються найкращими за точністю та швидкістю алгоритмами знаходження об'єктів на сцені. Починаючи з 2012 року, нейромережі займають перші місця на відомому міжнародному конкурсі з розпізнавання образів ImageNet.

Саме тому у роботі була використана згортова нейронна мережа, яка заснована на принципах неокогнітрону та доповнена навчанням за алгоритмом зворотнього розповсюдження помилки.

1.11 Постановка задачі дослідження

Тема дослідження емоційного стану дитини є актуальною, а головне актуальним є пошук або створення інструменту, який допоможе правильно виявити цей емоційний стан задля запобігання шкоди дитини, тобто виявлення психологічних проблем на ранньому етапі.

Тому, задачу кваліфікаційної роботи можна зафіксувати наступним чином: розробити безкоштовний додаток із відкритим вихідним кодом, який аналізує дитячі малюнки, для проведення первинної діагностики емоційного стану дитини. Також додаток може стати у нагоді безпосередньо для спеціалістів, які професійно діагностують стан дитини.

Потрібно побудувати нейромережеву модель аналізу дитячих малюнків для діагностики емоційного стану дитини. Для досягнення цієї мети в якості підзадачі можна виокремити аналіз і дослідження питання того, як

спеціалісти психологи аналізують дитячі малюнки у своїй повсякденній роботі відповідаючи на таке саме питання. Безумовно це першочергова необхідність, адже наша мета створити інструмент, який зможе якщо не замінити спеціаліста, то полегшити йому життя. Тобто нам важливо розуміти за якими ознаками малюнка можна визначити емоційний стан дитини.

Звичайно, важливим питанням при роботі з нейронними мережами є навчання цих самих моделей. Отже, наступною задачею стане підготовка датасету на якому буде навчатися нейромережа. Коли датасет буде підготовлений, наступною задачею стане навчання моделі. Потрібно дослідити алгоритм навчання нашої обраної моделі, навчити модель на підготовленому датасеті, та перевірити, що наша мета була досягнута – модель з високою відсотковою ймовірністю правильного результату на виході видає дані емоційного стану дитини проаналізувавши її малюнок.

2 СТВОРЕННЯ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ АНАЛІЗУ ДИТЯЧИХ МАЛЮНКІВ

2.1 Пошук умов для визначення емоційного стану за малюнком

Люди знаходять різні способи вираження своїх почуттів. Коли ми маленькі, ми не можемо дуже добре спілкуватися за допомогою слів і голосу, оскільки ми все ще вчимося писати і говорити, тому малюнки є особливим ресурсом для вираження наших емоцій і наших стосунків із сім'єю та навіть іншими речами у світі.

Малюнки дають нам чудове уявлення про дитячі страхи, радощі, кошмари, переживання, а також особистості. Малюнок містить всю ключову інформацію про них.

Це не остаточний спосіб пізнати дитину, давайте прийнемо це, інколи це нічого не означає, це лише результат розваги, але іноді малюнок може показати, що вони відчують і думають у цей момент.

Під час пошуку інформації за темою розшифрування дитячих малюнків, були знайдені аспекти та критерії за якими спеціалісти проводять свій аналіз щодо дитячих малюнків, а саме кольора, форми, розміри об'єктів, стиль тощо.

Проаналізувавши інформацію з інтернету за темою виявлення потенційних проблем у дитини дивлячись на її малюнок, стало зрозуміло що якщо психічне здоров'я дитини знаходиться в нормі, то в більшості випадків малюнок буде дійсно позитивний – багатокольоровий, з позитивними чи нейтральними об'єктами, такими як тварини, природа, сонце, небо, веселка тощо. Навіть якщо хлопчик намалює машину на дорозі (що характерно для хлопців), зазвичай на такому малюнку також буде сонце.

Інший важливий аспект, який психологу намагаються виявити в психічному стані дитини це жорстокість. Зазвичай малюнки дітей з таким

психічним відхиленням мають багато червоного кольору, намальовані об'єкти є неохайними, хаотично розмальовані, самі об'єкти також є фігурами що відображають ненависть та жорстокість.

Також можна виокремити тривожних дітей, в депресивному стані, які мало чим цікавляться, що є нетиповим для дитини. Діти можуть не відчувати належного догляду з боку батьків та почуватися самотньо. Характерними малюнками є багато хмар замість сонця, дощі, сірі відтінки кольорів.

2.2 Визначення емоційного стану за малюнком

Користуючись знайденою інформацією було вирішино виокремити три категорії емоцій, на які нейронна мережа буде декодувати дитяче зображення:

- щастя;
- тривога і депресія;
- гнів і насильство.

Малюнок може виражати багато почуттів, деякі почуття схожі, тому було вирішино об'єднати деякі з них в одній категорії.

Таким чином, до категорії «Гнів і насильство» будуть відноситись малюнки про війну, злими емоціями обличчя на малюнках або фігури, що виражають насильство (рис. 2.1).



Рисунок 2.1 – Приклад малюнків, що відповідають категорії «Гнів і насильство»

Як було зазначено раніше, характерними ознаками малюнків цієї категорії є хаотичне розмалювання, вогонь, переважність червоного кольору, контурні та неохайні об'єкти.

До категорії «Тривога і депресія» – малюнки з обличчями, що плачуть, пейзажі з дощем і хмарами, переважність сірих відтінків (рис. 2.2).



Рисунок 2.2 – Приклад малюнків, що відповідають категорії «Тривога і депресія»

Категорія «Щастя» – рисунки з домашніми тваринами, пейзажі, сонце, квіти, веселки тощо (рис. 2.3).



Рисунок 2.3 – Приклад малюнків, що відповідають категорії «Щастя»

Таким чином, на етапі навчання нейронної мережі буде підготовлено датасет із дитячих зображень, які відповідають нашим трьом категоріям.

2.3 Побудова нейромережевої моделі аналізу дитячих малюнків

Згортова нейронна мережа складається з шарів різних видів: згорткові (англ. convolutional) шари, субдискретизуючі (англ. subsampling, підвибірка) шари та шари «звичайної» нейронної мережі – персептрону, відповідно до рисунку 2.4.

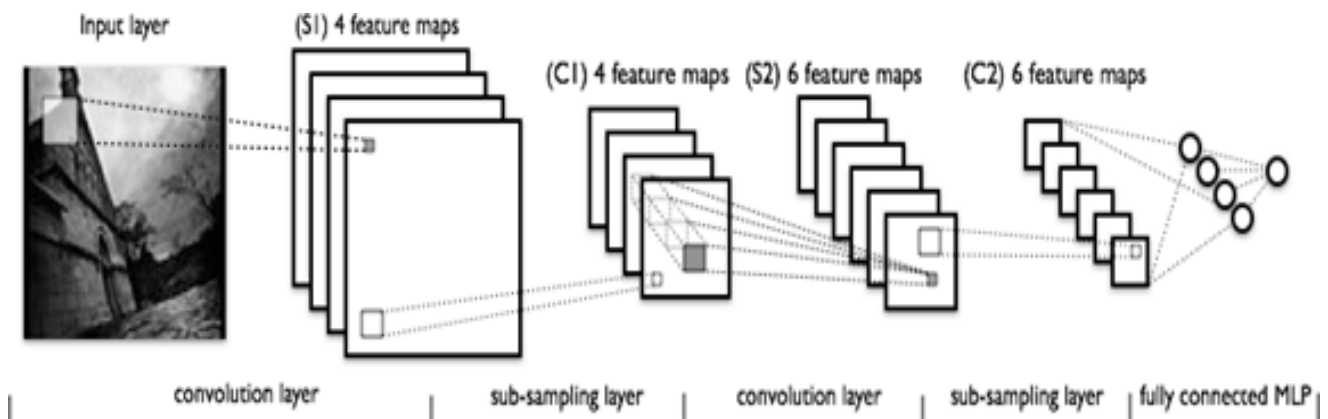


Рисунок 2.4 – Топологія згорткової нейронної мережі

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

Свою назву згортка отримала за назвою операції – згортка, суть якої буде описана далі.

Згорткові мережі є вдалою серединою між біологічно правдоподібними мережами та звичайним багатошаровим перцептроном [8]. На сьогоднішній день найкращі результати у розпізнаванні зображень отримують за їх допомогою. У середньому точність розпізнавання таких мереж перевищує стандартні штучні нейронні мережі на 10-15%. Згорткова нейронна мережа – це ключова технологія Deep Learning.

Візуалізація згортки та підвибірки показані на рисунку 2.5.

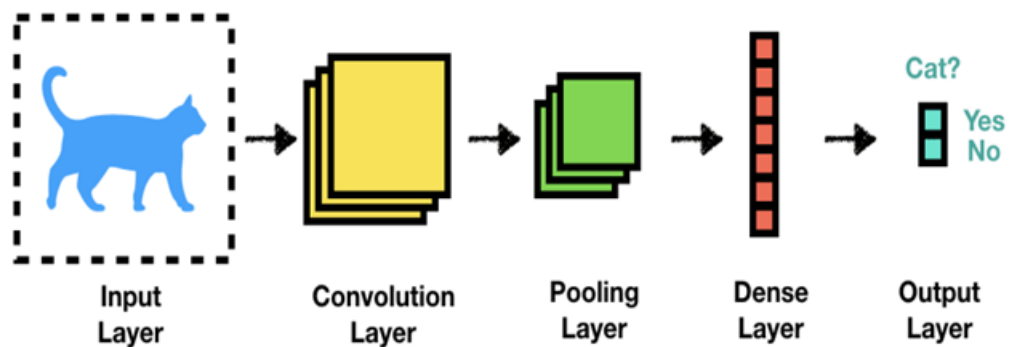


Рисунок 2.5 – Візуалізація згортки та підвибірки

Основною причиною успіху згорткових нейронних мережа стала концепція загальної ваги. Незважаючи на великий розмір, ці мережі мають невелику кількість параметрів, що налаштовуються в порівнянні з їх попередником – неокогнітроном. Є варіанти згорткових нейронних мереж (Tiled Convolutional Neural Network), схожих на неокогнітрон, в таких мережах відбувається, часткова відмова від пов'язаних ваг, але алгоритм навчання залишається тим самим і ґрунтується на зворотньому поширенні помилки. Мережі можуть швидко працювати на послідовній машині і швидко навчатися за рахунок прямого розпаралелювання процесу згортки по

кожній карті, а також зворотньої згортки при поширенні помилки по мережі (рис. 2.6).

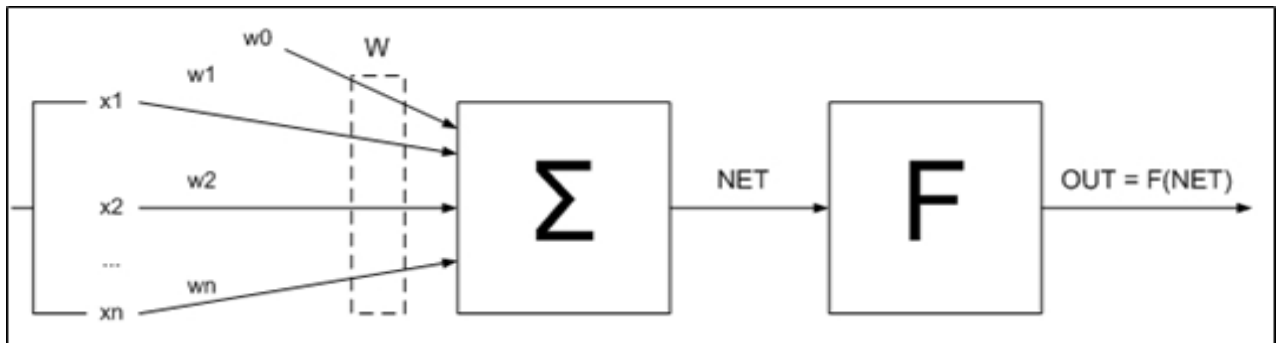


Рисунок 2.6 – Модель нейрона

Математична модель нейрона:

$$NET = \sum_{i=1}^n w_i * x_i + w_0, \quad (2.1)$$

де w_i – вага нейрона,

x_i – вихід нейрона,

w_0 – допоміжний параметр (зміщення),

w_i – вага i нейрона,

n – кількість синаптичних зв'язків, які входять в нейрон.

2.3.1 Топологія згорткової нейромережі

Визначення топології мережі орієнтується на завдання, що буде вирішуватись, дані з наукових статей та власний експериментальний досвід.

Можна виокремити такі етапи, що впливають на вибір топології:

- визначити вирішуване завдання нейромережею (класифікація, прогнозування, модифікація);
- визначити обмеження у розв'язуванні задачі (точність відповіді);
- визначити вхідні (тип: зображення, звук, розмір: 100x100, 30x30,

формат: RGB, в градаціях сірого) та вихідні дані (кількість класів).

Розв'язуване створюваною нейромережею завдання – класифікація зображень. Обмеження на мережу, що накладаються – це точність розпізнавання не менше 80%. Топологію зображено на рисунку 2.7.

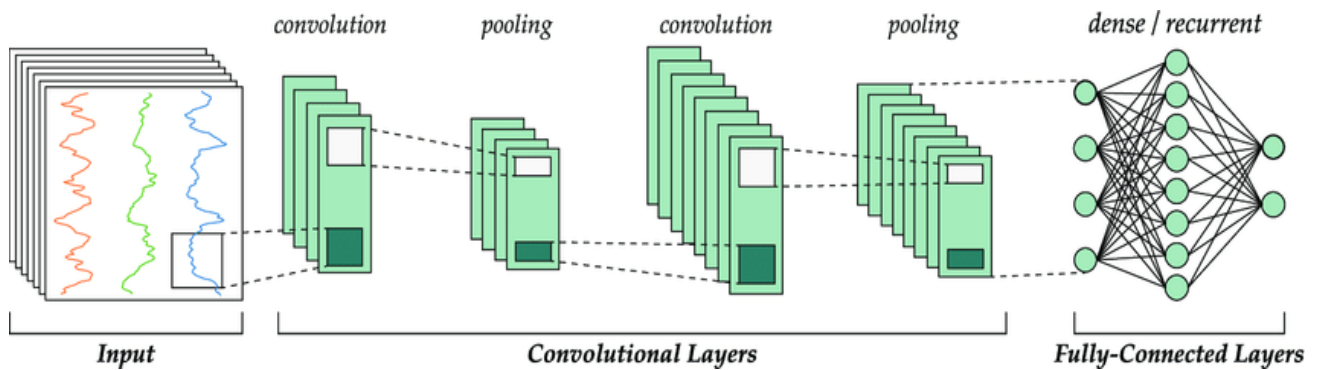


Рисунок 2.7 – Загальна топологія згорткової нейромережі

2.3.2 Вхідний шар

Вхідні дані самі по собі є кольоровими зображення типу JPEG, розміру 224x224 пікселів. Якщо розмір буде занадто великий, то обчислювальна складність підвищиться, відповідно швидкість відповіді буде довша, проте надається перевага точності розпізнавання. Якщо вибрати розмір занадто маленький, мережа не зможе виявити ключові ознаки. Кожне зображення розбивається на 3 канали: червоний, синій, зелений. Таким чином виходить 3 зображення розміру 224x224 пікселів.

Вхідний шар враховує двовимірну топологію зображень і складається з кількох карт (матриць), карта може бути одна, в тому випадку, якщо зображення представлене у відтінках сірого, інакше їх три, де кожна карта відповідає зображенню з конкретним каналом (червоним, синім і зеленим).

Вхідні дані кожного конкретного значення пікселя нормалізуються в діапазон від 0 до 1 за формулою:

$$f(p, \min, \max) = \frac{p - \min}{\max - \min}, \quad (2.2)$$

де f – функція нормалізації,

p – значення конкретного кольору пікселя від 0 до 255,

\min – мінімальне значення пікселя – 0,

\max – максимальне значення пікселя – 255.

2.3.3 Згортковий шар

Згортковий шар є набором карт (інша назва – карти ознак, в побуті це звичайні матриці), у кожній карті є синаптичне ядро (у різних джерелах його називають по-різному: скануюче ядро або фільтр).

Кількість карток визначається вимогами до завдання, якщо взяти велику кількість карток, то підвищиться якість розпізнавання, але збільшиться обчислювальна складність. Виходячи з аналізу наукових статей, у більшості випадків пропонується брати співвідношення один до двох, тобто кожна карта попереднього шару (наприклад, у першого згорткового шару, попереднім є вхідний) пов'язана з двома картами згорткового шару, відповідно до рисунку 2.8.

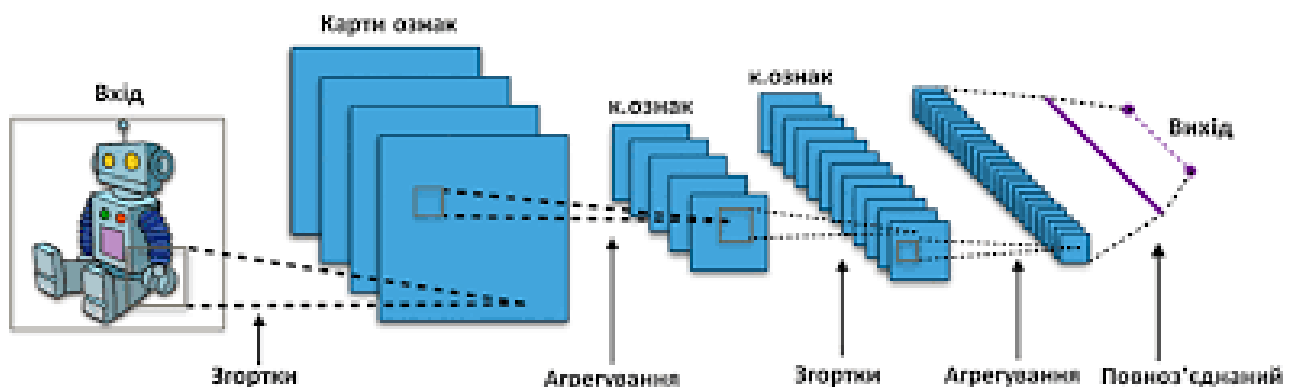


Рисунок 2.8 – Організація зв'язків між картами згорткового шару та

попереднього

Розмір у всіх карт згорткового шару – однакові та обчислюються за формулою:

$$(w, h) = (mW - kW + 1, mH - kH + 1), \quad (2.3)$$

де (w, h) – обчислювальний розмір згорткової карти,

mW – ширина попередньої карти,

mH – висота попередньої карти,

kW – ширина ядра,

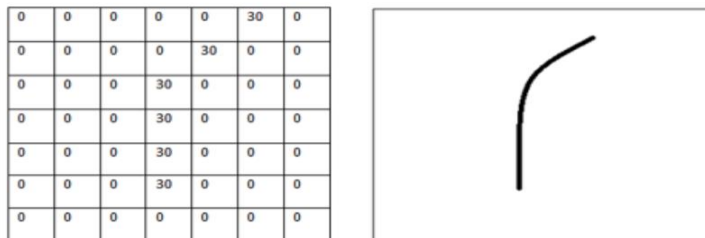
kH – висота ядра.

Ядро є фільтром або вікном, яке ковзає по всій області попередньої карти і знаходить певні ознаки об'єктів. Наприклад, якщо мережу навчали на безлічі дитячих малюнків із ландшафтом (найтиповіші дитячі малюнки з домішкою, травою, деревами, квітками, небом, сонцем тощо), то одне з ядер могло б у процесі навчання видавати найбільший сигнал при знаходженні подібних геометричних форм (коло з прямими – сонце, чи прямокутник з трикутником вгорі – будівля), інше ядро могло б виявляти інші ознаки, навіть кольори та їх насиченість. Найперші ядра працюють з найменшими ознаками як було розглянуто вище – такі як геометричні відповідності. Останні ж – навпаки можуть навіть вирішувати завдання детекції, тобто розпізнавати конкретні об'єкти на зображенні в процесі навчання. Розмір ядра зазвичай беруть у межах від 3x3 до 7x7. Якщо розмір ядра маленький, воно не зможе виділити будь-які ознаки, якщо занадто велике, то збільшується кількість зв'язків між нейронами. Також розмір ядра вибирається таким, щоб розмір карт згорткового шару був парним, це дозволяє не втрачати інформацію при зменшенні розмірності в підвибірковому шарі, описаному нижче [9].

Ядро являє собою систему ваг або синапсів, що розділяються, це одна з головних особливостей згорткової нейромережі. У звичайній багат шаровій мережі дуже багато зв'язків між нейронами, тобто синапсів, що дуже сповільнює процес детектування. У згортковій мережі – навпаки, загальні

ваги дозволяють зменшити кількість зв'язків і дозволити знаходити одну й ту саму ознаку по всій області зображення (рис. 2.9).

Ядро з навченою ознакою



Вхідне зображення

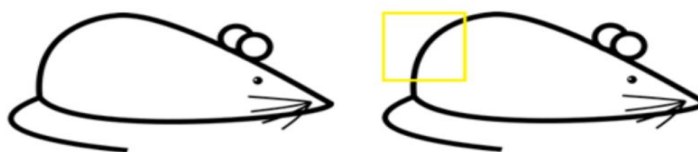


Рисунок 2.9 – Приклад ядра з навченою ознакою

У місці де жовте вікно буде високий рівень відгуку (сигналу), що буде означати наявність ознаки на вхідному зображенні.

Спочатку значення кожної карти згорткового шару дорівнюють 0. Значення важелів ядер задаються випадковим чином в області від -0.5 до 0.5. Ядро ковзає по попередній карті і здійснює операцію згортки, яка часто використовується для обробки зображень, що демонструє формула:

$$(f * g)[m,n] = \sum_{k,l} f[m - k, n - l] * g[k,l], \quad (2.4)$$

де f – початкова матриця зображення,

g – ядро згортки.

Неформально цю операцію можна описати так – вікном розміру ядра g проходимо із заданим кроком (зазвичай 1) усе зображення f , на кожному кроці поелементно множимо вміст вікна на ядро g , результат підсумовується і записується в матрицю результату, як на рисунку. 2.10.

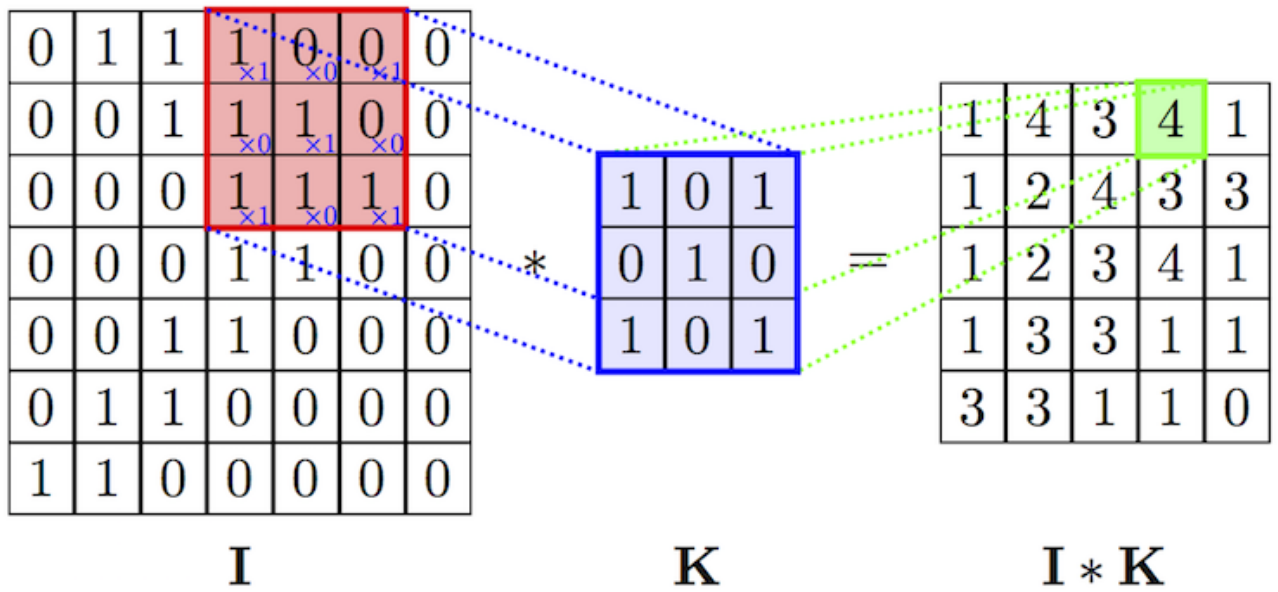


Рисунок 2.10 – Операція згортки та отримання значень згорткової картки

При цьому залежно від методу обробки меж вихідної матриці результат може бути меншим від вихідного зображення (valid), такого ж розміру (same) або більшого розміру (full), відповідно до рисунка 2.11.

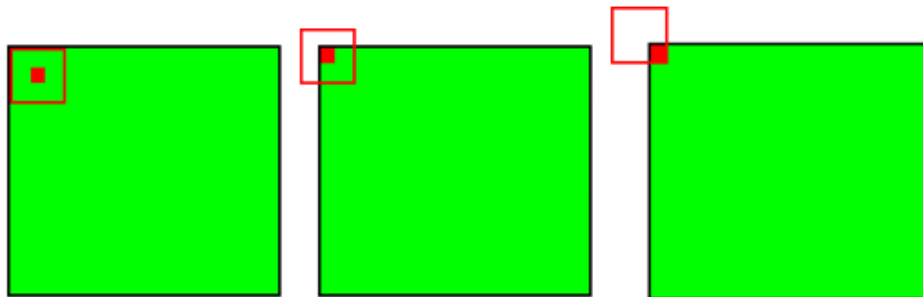


Рисунок 2.11 – Три види згортки вихідної матриці

У спрощеному вигляді цей шар можна описати формулою:

$$x^l = f(x^{(l-1)} * k^l + b^l), \quad (2.5)$$

де x^l – вихід шару l ,

$f()$ – функція активації,

b^l – коефіцієнт зсуву шару l ,

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

При цьому за рахунок крайових ефектів розмір вихідних матриць зменшується:

$$x_j^l = f(\sum_i x_i^{l-1} * k_j^l + b_j^l), \quad (2.6)$$

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

$f()$ – функція активації,

b^l – коефіцієнт зсуву шару l для ознак j ,

k_j^l – ядро згортки j карти слою l ,

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

2.3.4 Повнозв'язний шар

Останній з типів шарів – це шар звичайного багатошарового персептрона. Мета шару – класифікація, що моделює складну нелінійну функцію, оптимізуючи яку, покращується якість розпізнавання (рис. 2.12).

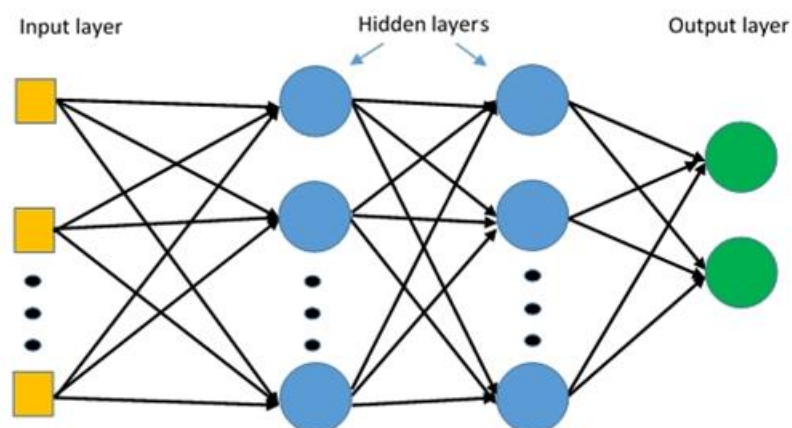


Рисунок 2.12 – Шар багат шарового персептронну

Нейрони кожної карти попереднього підвибіркового шару пов'язані з одним нейроном прихованого шару. Таким чином число нейронів прихованого шару дорівнює числу карт підвибіркового шару, але зв'язки можуть бути не обов'язково такими, наприклад, тільки частина нейронів будь-якої з карт підвибіркового шару може бути пов'язана з першим нейроном прихованого шару, а частина, що залишилася з другим, або всі нейрони першої карти пов'язані з нейронами першого та другого прихованого шару. Обчислення значень нейрона можна описати формулою:

$$x_j^l = f\left(\sum_i x_i^{l-1} * w_{i,j}^{l-1} + b_j^{l-1}\right), \quad (2.7)$$

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

$f()$ – функція активації,

b^l – коефіцієнт зсуву шару l для ознак j ,

$w_{i,j}^l$ – матриця вагових коефіцієнтів шару l .

2.3.5 Вихідний шар

Вихідний шар пов'язаний із усіма нейронами попереднього шару.

Кількість нейронів відповідає кількості класів, що розпізнаються, тобто 3:

«Щастя», «Тривога і депресія» та «Гнів і насильство».

Тобто результатом роботи згорткової нейронної мережі буде класифікація зображення до одного з цих трьох класів. Зазвичай, виходом є лише одне значення, а саме клас якому зображення відповідає з найбільшою вірогідністю. Проте специфікою завдання є те що необхідно знати усі вірогідності відношення певного малюнка до одного чи іншого класу, тож нейронна мережа буде повертати відсоткові вірогідності для усіх трьох класів (сума вірогідностей належання дитячого малюнка до кожного з трьох

класів буде дорівнювати одиниці).

2.3.6 Вибір функції активації

Одним із етапів розробки нейронної мережі є вибір функції активації нейронів. Вигляд функції активації багато в чому визначає функціональні можливості нейронної мережі та метод навчання цієї мережі. Класичний алгоритм зворотного поширення помилки добре працює на двошарових та тришарових нейронних мережах, але при подальшому збільшенні глибини починає відчувати проблеми. Однією з причин є так зване згасання градієнтів. У міру поширення помилки від вихідного шару до вхідного на кожному шарі відбувається збільшення поточного результату на похідну функції активації. Похідна у традиційної сигмоїдної функції активації менше одиниці по всій області визначення, тому після кількох шарів помилка стане близькою до нуля. Якщо ж, навпаки, функція активації має необмежену похідну (як, наприклад, гіперболічний тангенс), може статися вибухове збільшення помилки у міру поширення, що призведе до нестійкості процедури навчання.

У роботі як функція активації у прихованих та вихідному шарах застосовується гіперболічний тангенс, у згорткових шарах застосовується ReLU.

Використання гіперболічного тангенсу обумовлено наступними причинами:

- симетричні активаційні функції типу гіперболічного тангенсу забезпечують більш швидку збіжність, ніж стандартна логістична функція;
- функція має безперервну першу похідну;
- функція має просту похідну, яка може бути обчислена через її значення, що дає економію обчислень (рис. 2.13).

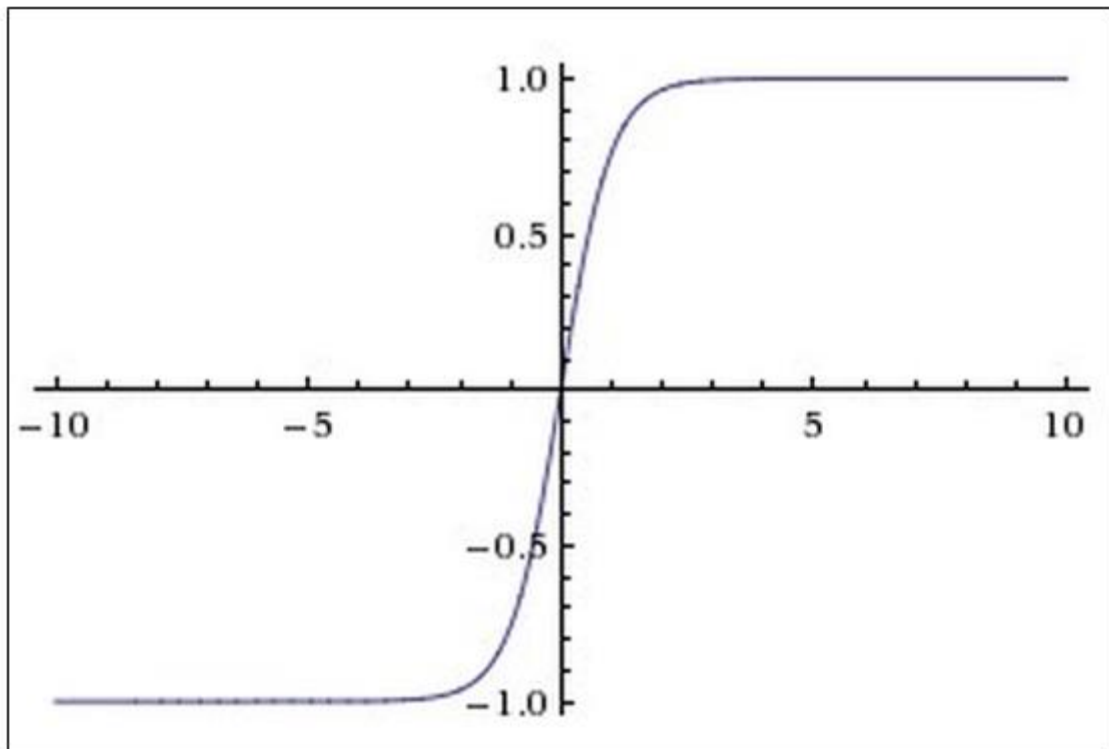


Рисунок 2.13 – Графік функції гіперболічного тангенсу

Функція гіперболічного тангенсу:

$$\tanh(z) = \frac{2}{(1+e^{(-2z)})} - 1 \quad (2.8)$$

Відомо, що нейронні мережі здатні наблизити як завгодно складну функцію, якщо в них достатньо шарів і функція активації є нелінійною. Функції активації на кшталт сигмоїдної чи тангенціальної є нелійними, але призводять до проблем із загасанням чи збільшенням градієнтів. Однак можна використовувати і набагато простіший варіант – випрямлену лінійну функцію активації (rectified linear unit, ReLU), яка виражається формулою:

$$f(s) = \max(0, s) \quad (2.9)$$

Графік функції ReLU зображено на рисунку 2.14.

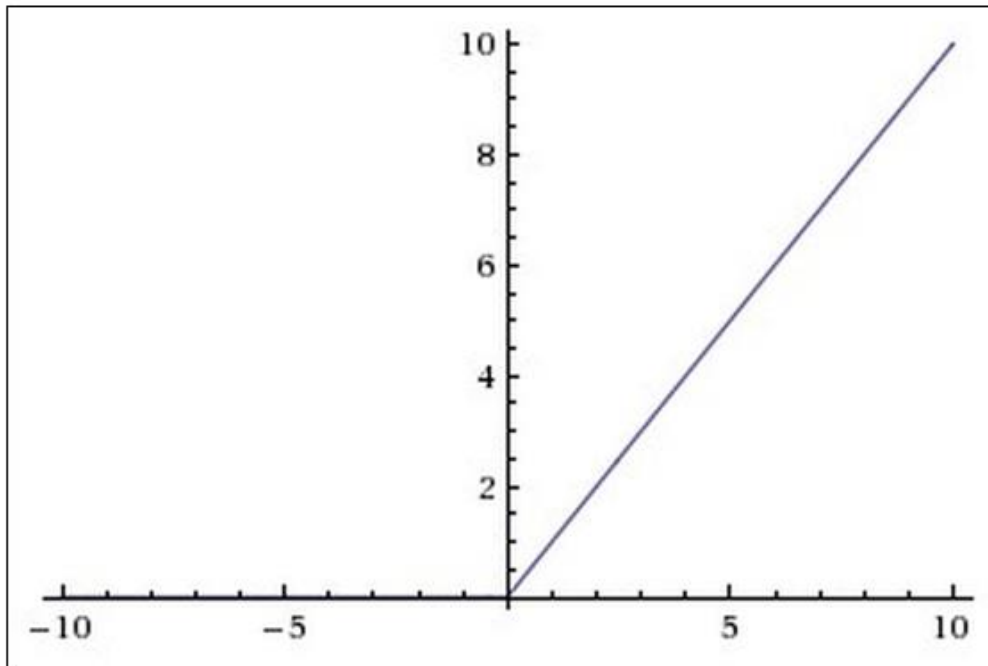


Рисунок 2.14 – Графік функції ReLU

Переваги використання ReLU:

- її похідна дорівнює або одиниці, або нулю, і тому може статися розростання чи згасання градієнтів, тому що помноживши одиницю на дельту помилки ми отримаємо дельту помилки, якщо ми б використовували іншу функцію, наприклад, гіперболічний тангенс, то дельта помилки могла, або зменшитися, або зрости, або залишитися такою ж, тобто, похідна гіперболічного тангенсу повертає число з різним знаком і величиною, що може відчутно вплинути на згасання чи розростання градієнта. Більш того, використання цієї функції призводить до проріджування ваг;

- обчислення сигмоїди та гіперболічного тангенсу вимагає виконання ресурсомістких операцій, таких як зведення у ступінь, у той час як ReLU може бути реалізований за допомогою простого порогового перетворення матриці активацій у нулі;

- відсікає непотрібні деталі у каналі при негативному виході.

З недоліків можна відзначити, що ReLU не завжди є достатньо надійною і в процесі навчання може виходити з ладу. Наприклад, великий

градієнт, що проходить через ReLU, може призвести до такого оновлення вагів, що даний нейрон ніколи більше не активується. Якщо це станеться, то, починаючи з цього моменту, градієнт, що проходить через цей нейрон, завжди дорівнюватиме нулю. Відповідно, даний нейрон буде незворотно виведений з ладу. Наприклад, при занадто великій швидкості навчання (learning rate) може виявитися, що до 40% ReLU «мертві» (тобто ніколи не активуються). Ця проблема вирішується шляхом вибору належної швидкості навчання.

2.4 Алгоритм машинного навчання

Згорткові нейронні мережі навчаються на наборі позначених зображень. Починаючи з заданого зображення, воно поширюється через різні шари CNN і повертається шуканий результат.

У цьому буде розглянуто детальніше алгоритм навчання з техніками, які використовуються для розширення даних.

2.4.1 Попередня обробка даних

Збільшення даних – це крок збільшення кількості зображень у певному наборі даних. Існує багато методів, які використовуються для збільшення даних, наприклад:

- відсікання;
- обертання (поворот зображення);
- гортання (зміна орієнтації зображення);
- ін'єкція шуму;
- трансформація колірного простору.

Це забезпечує краще навчання завдяки більшому розміру навчального набору та дозволяє алгоритму навчатися на різних станах об'єкта, про який йде мова.

Коли набір даних буде готовий, він ділиться на три частини, як будь-який проект машинного навчання:

- навчання набору: використовується для навчання алгоритму та створення пакетів;
- набір розробників: використовується для точного налаштування алгоритму та оцінки зміщення та дисперсії;
- тестовий набір: використовується для узагальнення помилки точності кінцевого алгоритму.

2.4.2 Пряме та зворотнє поширення

Згорткові нейронні мережі – це особливий вид нейронних мереж, що спеціалізується на зображеннях. Навчання в нейронних мережах – це етап обчислення ваг параметрів, визначених вище, у кількох рівнях.

Іншими словами, метою є знаходження найкращих параметрів, які дають найкращий прогноз/наближення y_i починаючи з вхідного зображення x_i , реального значення y_i . Для цього ми визначаємо цільову функцію, яка називається функцією втрат і позначається J , яка кількісно визначає відстань між реальними та прогнозованими значеннями на загальному навчальному наборі. Мінімізуємо J , виконавши два основні кроки:

- попереднє розповсюдження: ми поширюємо дані через мережу або повністю, або пакетами, і обчислюємо функцію втрат для цього пакета, яка є нічим іншим, як сумою помилок, допущених на прогнозованому виході для різних рядків;
- зворотнє поширення: полягає в обчисленні градієнтів функції вартості відносно різних параметрів, а потім застосуванні алгоритму спаду для їх оновлення.

Повторюємо той самий процес кілька разів, який називається номером епохи. Після визначення архітектури алгоритм навчання записується наступним чином:

- Ініціалізація параметрів моделі, крок, еквівалентний введенню шуму в модель;

- For $i=1,2\dots N$: (N – кількість епох)

Останній крок включає в себе виконання прямого поширення. $\forall i$ обчислює прогнозоване значення x_i через нейронну мережу y_i^θ ; оцінка функції за формулою:

$$J(\theta) = \frac{1}{m} \sum L(y_i^\theta, y_i), \quad (2.10)$$

де m – це розмір навчального сету,

θ – параметри моделі,

L – функція витрат (функція витрат оцінює відстань між реальним і прогнозованим значенням в одній точці).

Також функцію зворотнього поширення, що застосовує метод зниження для оновлення параметрів:

$$\theta = G(\theta). \quad (2.11)$$

2.4.3 Архітектура ResNet

ResNet, коротке з'єднання або з'єднання з пропуском – це згортковий рівень n , який враховує рівень $n-2$. Інтуїція впливає з того факту, що коли нейронні мережі стають дуже глибокими, точність на виході стає дуже стабільною і не зростає. Введення залишків з попереднього шару допомагає вирішити цю проблему (рис. 2.15).

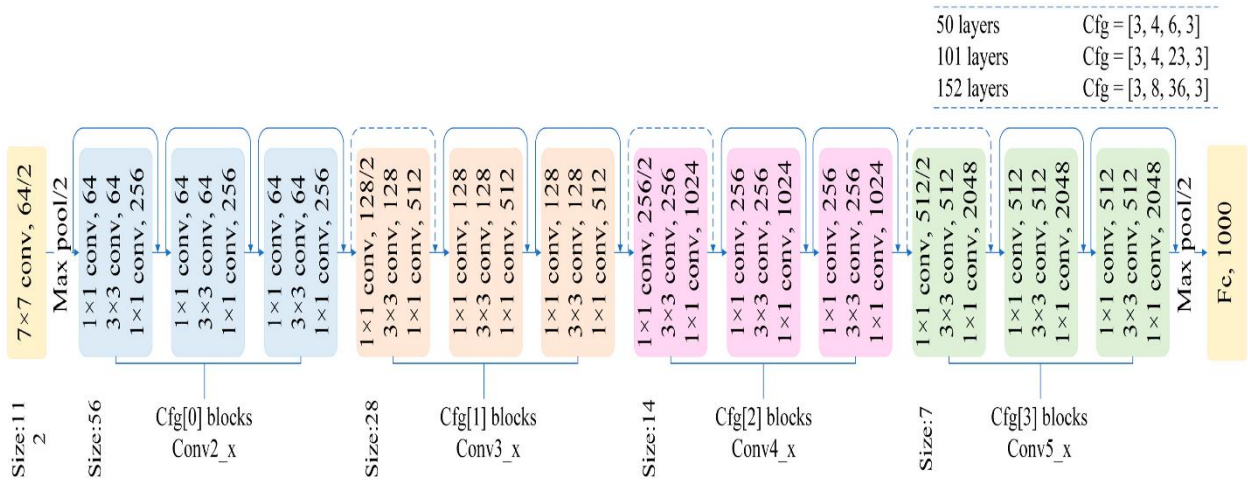


Рисунок 2.15 – Архітектура ResNet

Розглянемо залишковий блок i , коли з'єднання пропуску вимкнено, отримаємо таке рівняння:

$$z_j^{[i]} = \sum_{l=1}^{n^{(i-1)}} w_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]} \rightarrow a_l^{[i]} = \psi^{[i]}(z_j^{[i]}). \quad (2.12)$$

У випадку, коли ввімкнено пропуск з'єднання, маємо:

$$z_j^{[i]} = \sum_{l=1}^{n^{(i-1)}} w_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]} \rightarrow a_l^{[i]} = \psi^{[i]}(z_j^{[i]} + a_j^{[i-2]}). \quad (2.13)$$

При правильному виборі функції активації $\psi^{[i]}$ і $w^{[i]} = 0$, $b^{[i]} = 0$, отримаємо:

$$a^{[i]} = a^{[i-2]}. \quad (2.14)$$

Залишковий блок здатний вивчати ідентифікаційну функцію i , таким чином, не завдає шкоди нейронній мережі [11]. Зазвичай потрібно, щоб $a^{[i]}$, та $a^{[i-2]}$, мали однакову форму, тому часто використовується однакова згортка. Якщо ні, встановлюється:

$$\rightarrow a^{[i]} = \psi^{[i]}(z^{[i]} + W_s a^{[i-2]}) \dim(W_s) = [n^{[i]}, n^{[i-2]}], \quad (2.15)$$

де W_s може бути фіксованим або вивченим тензором. Підсумок залишкового блоку зображено на рисунку. 2.16.

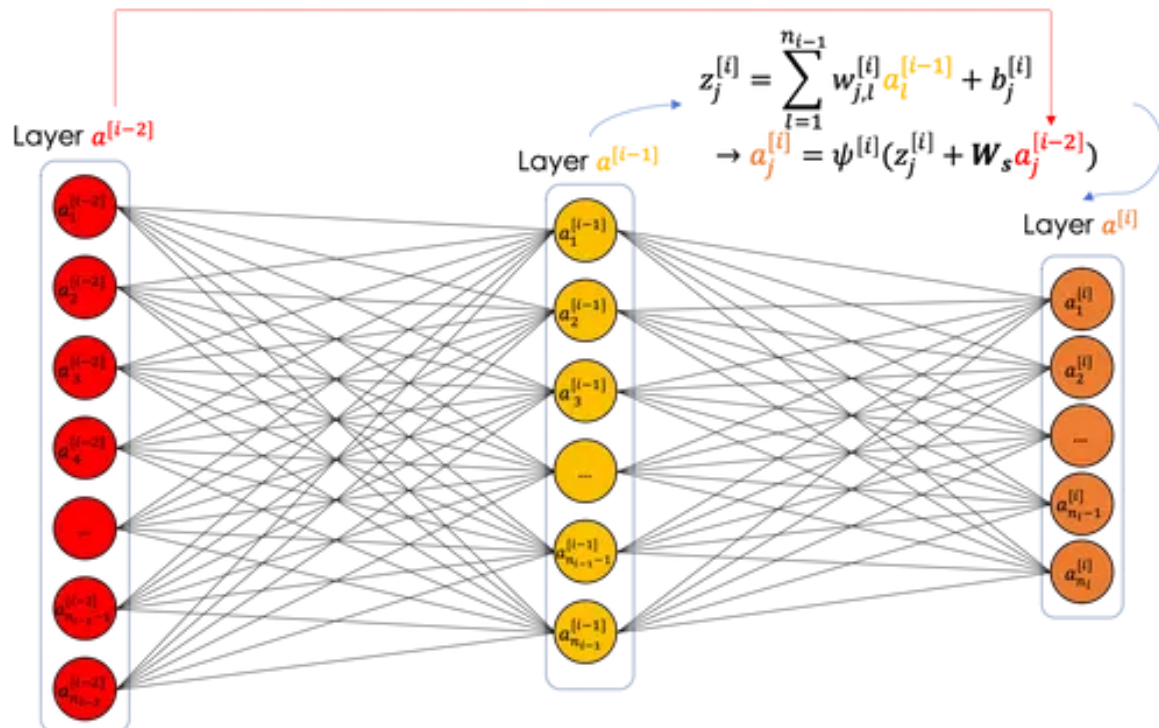


Рисунок 2.16 – Залишковий блок

2.4.4 Навчання згорткової нейронної мережі

Останнім етапом залишається власне навчання нейронної мережі. Навчальна вибірка складається з трьох категорій. Як було зазначено раніше, маємо 3 категорії «Щастя», «Гнів і насильство» та «Тривога і депресія». Звичайно, дуже важливим фактором є вибірка на якій мережа буде навчатися. З першою категорією не виникає питань: тотально більша частина дитячих малюнків що можна знайти в мережі підпадають саме під першу категорію «Щастя». Проте в решті решт була знайдена певна кількість інших зображень, що підходять до залишившихся двох категорій.

На початковому етапі нейронна мережа є ненавченою (неналаштованою). У загальному сенсі під навчанням розуміють послідовний

показ образу на вхід нейромережі, з навчального набору, потім отримана відповідь порівнюється з бажаним виходом, у нашому випадку це вирогідність по всім трьом класам, отримана різниця між очікуваною та отриманою відповіддю є результат функції помилки (дельта помилки). Потім цю дельту помилки необхідно поширити на пов'язані нейрони мережі [12].

Таким чином, навчання нейронної мережі зводиться до мінімізації функції помилки, шляхом коригування вагових коефіцієнтів синаптичних зв'язків між нейронами. Під функцією помилки розуміється різниця між отриманою відповіддю та бажаною. Наприклад, на вхід було подане зображення ландшафту літньої пори року (сонце, трава, небо, квітки тощо – очевидний клас «Щастя»), припустимо, що вихід нейромережі був 0.6, а бажаний результат 1 (знову ж таки тому що це впевнений клас «Щастя»), отримаємо помилку мережі різницею, тобто 0.4. Потім ваги вихідного шару нейронів коригуються відповідно до помилки. Для нейронів вихідного шару відомі їх фактичні та бажані значення виходів. Тому налаштування вагів зв'язків для таких нейронів є відносно простим.

Використавши попередньо навчену модель і PyTorch ми розділили набір даних 80:20, навчання та перевірку відповідно. Після кількох ітерацій і налаштування моделі, вона добре працювала з набором даних, який було створено.

3 ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Вибір тестових даних

Як було зазначено в минулих розділах, на виході з нейронної мережі очікується три класи з відсотковими вираженнями приналежності того чи іншого дитячого малюнку до цих класів. Отже, для навчання потрібен датасет, що містить у собі дитячі рисунки, які відповідають трьом категоріям: «Щастя», «Гнів і насильство» та «Тривога і депресія».

Насправді важко знайти багато матеріалу у мережі (до речі, було розглянуто потенційну складність пошуку задовільних матеріалів при створенні датасетів власноруч у розділі про недоліки нейромережей у цілому), проте в решті решт датасет було складено. Розглянемо кілька прикладів дитячих малюнків, які було використано при навчанні моделі на рисунках 3.1 та 3.2:



Рисунок 3.1 – Дитячий малюнок використаний для навчання нейронної мережі класу «Щастя»



Рисунок 3.2 – Дитячий малюнок використаний для навчання нейронної мережі класу «Щастя»

Бачимо на рисунках веселі обличчя з посмішками, яскраві та насичені кольори, позитивний природний ландшафт: трава, сонце, квіти тощо. Саме на таких типових зображеннях навчатиметься нейронна мережа. Насправді все вище перераховане стане непоганим гачком, за який мережа зможе зачіпитись при вирішенні задачі класифікації рисунка до того чи іншого класу вже після процесу навчання. Тобто, після навчання на таких зображеннях, високо зростає вірогідність того, що наступні зображення, що подаватимуться на вхід, які будуть мати схожі ознаки з цими рисунками, наприклад наявність сонця, чи щасливих лиць тощо буду класифіковані нейронною мережою до класу «Щастя».

Наступними для датасету були відібрані рисунки, що належать до

класу «Гнів і насильство» (рис. 3.3).



Рисунок 3.3 – Дитячий малюнок використаний для навчання нейронної мережі класу «Гнів і насильство»

У цьому рисунку характерною ознакою приналежності до класу «Гнів і насильство» є хаотичне та неохайне забарвлення рисунку. Рисунок майже повністю розмальований прямими «рисками». Загальні тона рисунка не можна назвати позитивними, кольора достатньо темні. Невисока різноманітність кольорів також є характерною ознакою для цього класу. Ключовим моментом є також полум'я на даху та зброя (важко розпізнати це навіть дорослій людині, тож звичайно складність розпізнання об'єктів нейромережею саме на дитячих малюнках є потенційною проблемою для

точності вихідних даних з мережі).

Ще один рисунок, застосований при навчанні моделі цього класу (рис.3.4).



Рисунок 3.4 – Дитячий малюнок використаний для навчання нейронної мережі класу «Гнів і насильство»

Аналогічна ситуація: у цілому зображення війни, сірі кольори, зброя, постріли, червоний колір полум'я.

Залишається останній клас «Тривога і депресія». Як було розглянуто раніше, для цього датасету потрібні рисунки з похмурими обличчями, хмарами, дощем, сірими кольорами. Є деякі збіжності з попереднім класом «Гнів і насильство», але все ж таки можливо виокремити специфічні ознаки цього класу.

Знайти досить велику кількість таких малюнків також було не легким завданням, адже більша частина дитячих малюнків у мережі мають позитивний характер, що досить логічно для дітей. Отже кілька прикладів

рисуноків, що були відібрані для навчання нейронної мережі здатності відрізняти клас «Тривога і депресія» (рис. 3.5).



Рисунок 3.5 – Дитячий малюнок використаний для навчання нейронної мережі класу «Тривога і депресія»

Родина з похмурими посмішками ще один непоганий кандидат для датасету цього класу – одразу відчувається депресивний настрій малюнку (рис. 3.6).



Рисунок 3.6 – Дитячий малюнок використаний для навчання нейронної мережі класу «Тривога і депресія»

3.2 Створення і тренування моделі

Спочатку необхідно імпортувати необхідні пакети:

1. `os` — читання файлів та структури директорій;
2. `numpy` — для деяких матричних операцій поза TensorFlow;
3. `matplotlib.pyplot` — побудова графіків та відображення зображень із тестового та валідаційного набору даних.

Розробку нашого класифікатора ми починаємо із завантаження набору даних. Набір даних є підготовлений датасет, що було розглянуто у минулому підрозділі. Його для зручності було перенесено всередину головного репозиторія в заздалегідь створену папку Data (рис. 3.7).

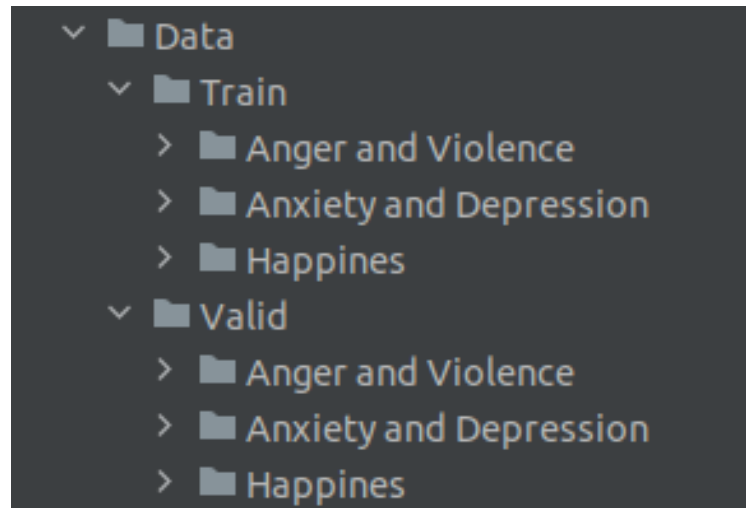


Рисунок 3.7 – Структура каталогу датасету для навчання моделі

Тепер потрібно завантажити цю структуру папок в Python програму, щоб мати доступ до цього каталогу на рівні коду для завантаження набору даних у нейронну мережу для її навчання (рис. 3.8).

```

from pathlib import Path
path = os.path.dirname(os.path.abspath(__file__))
ccc = Path.joinpath(Path(path).parent.absolute(), 'Data')
data_dir = '/Data'
  
```

Рисунок 3.8 – Код завантаження структури папок датасету у Python програму

Наступним етапом є трансформування вхідних даних, тобто потрібно зробити певні перетворення з вхідними картинками з датасету до того як вони потраплять на вхід до нейронної мережі (рис. 3.9).

```

data_transforms = {
    'Train': transforms.Compose([
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ]),
    'Valid': transforms.Compose([
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ]),
}

```

Рисунок 3.9 – Перетворення даних перед входом до нейронної мережі

Кілька пояснень щодо прикладу коду на рис. 3.9:

- `transforms.RandomResizedCrop(224)`: обрізає випадкову частину зображення та змінює її розмір до заданого розміру (224x224 пікселів);
- `transforms.RandomHorizontalFlip()`: горизонтально перегортає задане зображення випадковим чином (за замовчуванням ймовірність того, що зображення буде перегорнуте, або залишиться в первинному стані дорівнює 50 відсотків);
- `transforms.ToTensor()`: перетворює зображення на тензор.
- `transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])`: нормалізує тензорне зображення із середнім і стандартним відхиленням.

Далі створюємо об'єкт даталоадеру, який поєднує набір даних і вибірку, а також забезпечує ітерацію над заданим набором даних. `DataLoader` підтримує набори даних як у стилі карт, так і в ітераційному стилі з одно- або багатопроцесним завантаженням, налаштуванням порядку завантаження та додатковим автоматичним пакетуванням (зіставленням) і закріпленням пам'яті. Встановимо параметр `shuffle=True`, для того, щоб перетасувати

зображення в датасеті; `batch_size=4`, тобто зображення будуть подаватися у групі по 4 зображення; `num_workers=2`, для швидкості опрацювання (паралельно будуть опрацьовуватись 2 групи зображень) (рис. 3.10).

```
# Using the image datasets and the transforms, define the dataloaders
dataloaders = {x: torch.utils.data.DataLoader(image_datasets[x], batch_size=4,
                                             shuffle=True, num_workers=2)
               for x in ['Train', 'Valid']}
```

Рисунок 3.10 – Створення об'єкту DataLoader.

Усі ці перетворення здебільшого потрібні для того, щоб можна було дійсно переконатися, що нейронна мережа навчилася на заданому датасеті, а не просто запам'ятала зображення датасету.

Коли зображення готові на вхід в мережу, створюємо нейронну мережу. Як було розглянуто раніше, використана попередньо навчена нейронна мережа з архітектурою ResNet152, адже нам нема сенсу навчати модель з нуля, потрібно замінити лише останні шари нейронної мережі, перші будуть залишатися незмінними (рис. 3.11).

```
model = models.resnet152(pretrained=True)
classifier = nn.Sequential(OrderedDict([
    ('fc1', nn.Linear(2048, 1024)),
    ('relu', nn.ReLU()),
    ('fc2', nn.Linear(1024, 3)),
    ('output', nn.LogSoftmax(dim=1))
]))
model.fc = classifier
logging.info('Model was created')
```

Рисунок 3.11 – Створення моделі ResNet152.

Було використано `nn.Sequential` – послідовний контейнер. Модулі додаватимуться до нього в тому порядку, в якому вони передані в конструкторі або простому типу даних мови Python, як у нашому випадку, `OrderedDict` з модулями. `NN.Linear` – застосовує лінійне перетворення до вхідних даних. Перший параметр це розмір кожної вхідної вибірки, другий – розмір кожного вихідного зразка. Вихід з `fc1` рис.3.11 є входом до `nn.ReLU`, вихід з `nn.ReLU` є входом до `fc2`, відповідно вихід з `fc2` є входом до останнього модулю `nn.LogSoftMax`. Таким чином було замінено останні шари мережі та створено її фінальну версію.

Наступний етап це навчання. Воно буде відбуватися в 10 епох, в кожній з яких присутні етап навчання та перевірки (валідації). Під час тренування призначаємо режим роботи моделі: `train mode` та `eval mode` в залежності якщо це тренувальний чи валідаційний набір датасету відповідно (рис. 3.12).

```

for epoch in range(1, num_epochs+1):
    # Each epoch has a training and validation phase
    for phase in ['Train', 'Valid']:
        if phase == 'Train':
            scheduler.step()
            model.train() # Set model to training mode
        else:
            model.eval() # Set model to evaluate mode

```

Рисунок 3.12 – Встановлення режиму моделі в залежності від датасету.

Це змінює поведінку деяких шарів для поліпшення результатів. Наприклад `nn.Dropout` більше не втрачатиметься, а шари `nn.BatchNorm` будуть використовувати поточні оцінки замість пакетної статистики.

Далі ітеруємося по датасету, завантаженому в пам'ять. Рядок коду `torch.set_grad_enabled` очищає проміжні значення для оцінки, які необхідні

для зворотного поширення під час навчання, таким чином зберігаючи пам'ять. Його можна порівняти з оператором `with torch.no_grad()`, але приймає логічне значення (рис. 3.13).

```
# Iterate over data.
for inputs, labels in dataloaders[phase]:
    inputs, labels = inputs.to(device), labels.to(device)
    # zero the parameter gradients
    optimizer.zero_grad()

    # forward
    # track history if only in train
    with torch.set_grad_enabled(phase == 'Train'):
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        _, preds = torch.max(outputs, 1)
        # backward + optimize only if in training phase
        if phase == 'Train':
            loss.backward()
            optimizer.step()
```

Рисунок 3.13 – Ітерація даними під час навчання моделі.

У PyTorch параметри, які можна вивчати (тобто ваги та зміщення) моделі `torch.nn.Module`, містяться в параметрах моделі (до яких можна отримати доступ за допомогою `model.parameters()`). `State_dict` — це просто об'єкт словника Python, який відображає кожен шар на його тензор параметрів. Лише шари з параметрами, які можна вивчати (згорткові шари, лінійні шари тощо) і зареєстровані буфери (`batchnorm's running_mean`) мають записи в `state_dict` моделі. Об'єкти оптимізатора (`torch.optim`) також мають `state_dict`, який містить інформацію про стан оптимізатора, а також використані гіперпараметри.

Оскільки об'єкти `state_dict` є словниками Python, їх можна легко зберігати, оновлювати, змінювати та відновлювати, додаючи велику модульність моделям і оптимізаторам PyTorch (рис. 3.14).

```

optimizer.step()

# statistics
running_loss += loss.item() * inputs.size(0)
running_corrects += torch.sum(preds == labels.data)
epoch_acc = running_corrects.double() / dataset_sizes[phase]
# deep copy the model
if phase == 'Valid' and epoch_acc > best_acc:
    best_acc = epoch_acc
    best_model_wts = copy.deepcopy(model.state_dict())
model.load_state_dict(best_model_wts)

```

Рисунок 3.14 – Використання функції state_dict.

Можна побачити код, що збирає статистику під час кожної епохи. Таким чином, є можливість дізнатися точність навчання. Для підготовленого датасету точність класифікації досягла 87% (рис. 3.15).

```

def train_model(model, criterion, optimizer, scheduler, num_epochs=10):
    best_model_wts = copy.deepcopy(model.state_dict())
    best_acc = 0.0
    for epoch in range(1, num_epochs+1):
        # Each epoch has a training and validation phase
        for phase in ['Train', 'Valid']:
            if phase == 'Train':
                scheduler.step()
                model.train() # Set model to training mode
            else:
                model.eval() # Set model to evaluate mode
            running_loss = 0.0
            running_corrects = 0
            # Iterate over data.
            for inputs, labels in dataloaders[phase]:
                inputs, labels = inputs.to(device), labels.to(device)
                # zero the parameter gradients
                optimizer.zero_grad()
                # forward
                # track history if only in train
                with torch.set_grad_enabled(phase == 'Train'):
                    outputs = model(inputs)
                    loss = criterion(outputs, labels)
                    _, preds = torch.max(outputs, 1)
                # backward + optimize only if in training phase
                if phase == 'Train':
                    loss.backward()
                    optimizer.step()
                # statistics
                running_loss += loss.item() * inputs.size(0)
                running_corrects += torch.sum(preds == labels.data)
            epoch_acc = running_corrects.double() / dataset_sizes[phase]
            # deep copy the model
            if phase == 'Valid' and epoch_acc > best_acc:
                best_acc = epoch_acc
                best_model_wts = copy.deepcopy(model.state_dict())
    model.load_state_dict(best_model_wts)

```

Рисунок 3.15 – Повний код функції навчання моделі.

Останній крок це збереження навченої моделі. Для цього було використано функцію `torch.save`, яка зберігає серіалізований об'єкт на диск (утиліта для серіалізації `pickle` Python). За допомогою цієї функції можна зберігати моделі, тензори та словники всіх видів об'єктів (рис. 3.16).

```
checkpoint = {'input_size': [3, 224, 224],
             'batch_size': dataloaders['Train'].batch_size,
             'output_size': 3,
             'state_dict': model.state_dict(),
             'data_transforms': data_transforms,
             'optimizer_dict': optimizer.state_dict(),
             'class_to_idx': model.class_to_idx,
             'epoch': model.epochs}
torch.save(checkpoint, 'emotions80_checkpoint.pth')
```

Рисунок 3.16 – Збереження навченої моделі на диск.

Можна побачити підготовку всіх параметрів, що потрібні для використання функції `torch.save`: вказується розмір групи, що встановлювалась на етапі підготовки даних до навчання; класи за якими необхідно провести класифікацію (3 класи емоцій, що були визначені для класифікації дитячих зображень); кількість епох; дані функції `state_dict`, що була розглянута вище; розмір вхідних даних (зображення 224x224 пікселів, результат передопрацювання даних датасету), тощо.

3.3 Класифікація зображень навченою нейронною мережею

Після того, як мережа створена, навчена та збережена, можна використовувати її для класифікації будь-яких дитячих малюнків, поза межами навчального датасету. Але вхідне зображення також потребує

деякого передопрацювання (рис. 3.17).

```
def process_image(image):
    size = 361, 361
    image.thumbnail(size, Image.ANTIALIAS)
    image = image.crop((128 - 112, 128 - 112, 128 + 112, 128 + 112))
    npImage = np.array(image)
    npImage = npImage/255.

    imgA = npImage[:, :, 0]
    imgB = npImage[:, :, 1]
    imgC = npImage[:, :, 2]

    imgA = (imgA - 0.485)/(0.229)
    imgB = (imgB - 0.456)/(0.224)
    imgC = (imgC - 0.406)/(0.225)

    npImage[:, :, 0] = imgA
    npImage[:, :, 1] = imgB
    npImage[:, :, 2] = imgC

    npImage = np.transpose(npImage, (2, 0, 1))

    return npImage
```

Рисунок 3.17 – Функція передопрацювання вхідного зображення для класифікації.

Головна мета цього коду це перетворити зображення щоб воно мало розмір 224x224 пікселя. Використовується функція `Image.thumbnail`, яка перетворює зображення на мініатюру. Цей метод змінює зображення таким чином, щоб воно містило саму мініатюру, розмір якої не перевищує вказаний розмір. Цей метод обчислює відповідний розмір мініатюри, щоб зберегти аспект зображення, викликає метод `draft` для налаштування програми читання файлів (де це можливо) і, нарешті, змінює розмір зображення.

За допомогою бібліотеки `numpy` зображення розкладається на матрицю. За допомогою `numpy.transpose` можна виконувати просту функцію транспонування в межах одного рядка за допомогою методу `numpy.transpose()` `Numpy`. Він може транспонувати 2-D масиви, але не впливати на 1-D масиви. Цей метод транспонує 2-D масив `numpy`. Таким

чином результатом опрацювання зображення функцією `process_image`, на виході отримуємо матрицю цього зображення (рис. 3.18).

```
(Pdb) npImage
array([[ 2.2489083,  2.2489083,  2.2489083, ..., -2.11790393,
        -2.11790393, -2.11790393],
       [ 2.2489083,  2.2489083,  2.2489083, ..., -2.11790393,
        -2.11790393, -2.11790393],
       [ 2.2489083,  2.2489083,  2.2489083, ..., -2.11790393,
        -2.11790393, -2.11790393],
       ...,
       [ 2.2489083,  2.2489083,  2.2489083, ..., -2.11790393,
        -2.11790393, -2.11790393],
       [ 2.2489083,  2.2489083,  2.2489083, ..., -2.11790393,
        -2.11790393, -2.11790393],
       [ 2.2489083,  2.2489083,  2.2489083, ..., -2.11790393,
        -2.11790393, -2.11790393],
       [ 2.2489083,  2.2489083,  2.2489083, ..., -2.11790393,
        -2.11790393, -2.11790393],
       [ 2.42857143,  2.42857143,  2.42857143, ..., -2.03571429,
        -2.03571429, -2.03571429],
       [ 2.42857143,  2.42857143,  2.42857143, ..., -2.03571429,
        -2.03571429, -2.03571429],
       [ 2.42857143,  2.42857143,  2.42857143, ..., -2.03571429,
        -2.03571429, -2.03571429],
       ...,
       [ 2.42857143,  2.42857143,  2.42857143, ..., -2.03571429,
        -2.03571429, -2.03571429],
       [ 2.42857143,  2.42857143,  2.42857143, ..., -2.03571429,
        -2.03571429, -2.03571429],
       [ 2.42857143,  2.42857143,  2.42857143, ..., -2.03571429,
        -2.03571429, -2.03571429],
       [ 2.42857143,  2.42857143,  2.42857143, ..., -2.03571429,
        -2.03571429, -2.03571429],
       [ 2.64,  2.64,  2.64, ..., -1.80444444,
        -1.80444444, -1.80444444],
       [ 2.64,  2.64,  2.64, ..., -1.80444444,
        -1.80444444, -1.80444444],
       [ 2.64,  2.64,  2.64, ..., -1.80444444,
        -1.80444444, -1.80444444],
       ...,
       [ 2.64,  2.64,  2.64, ..., -1.80444444,
```

Рисунок 3.18 – Результат роботи функції `image`.

Після розкладання зображення на матрицю, воно готове потрапити на вхід до функції `torch.FloatTensor`, функції, що повертає об'єкт багатовимірної матриці, що містить елементи одного типу даних. Тобто функція `process_image` як раз була необхідна для виклику `torch.FloatTensor` та отримання тензора для подальшої роботи з ним.

Далі завантажується вже створена модель у програму, та викликається метод `forward` (в цей метод як параметр очікує тензор, що був створений за допомогою `torch.FloatTensor`), який опрацьовує матрицю зображення. Ці дані слугують входом до функції `torch.exp`, яка повертає новий тензор із експонентою елементів вхідного тензора. На цьому етапі нейромережева модель вже відпрацювала, залишається підготувати зрозумілі результати, що будуть вертатися із функції.

Таким чином модель повертає результати класифікації (рис. 3.18).

```
def predict(image_path, model, topk=5):  
    urllib.request.urlretrieve(image_path, "sample.png")  
    image = torch.FloatTensor([process_image(Image.open("sample.png"))])  
    model.eval()  
    output = model.forward(torch.autograd.Variable(image))  
    probabilities = torch.exp(output).data.numpy()[0]  
  
    top_idx = np.argsort(probabilities)[-topk:][::-1]  
    top_class = [idx_to_class[x] for x in top_idx]  
    top_probability = probabilities[top_idx]  
    return top_probability, top_class
```

Рисунок 3.18 – Функція predict, що повертає результати класифікації.

3.4 Результати роботи програми

Коли модель побудована і навчена, її можна використовувати для аналізу дитячих емоцій на зображеннях. Розглянемо результат роботи нейронної мережі на дитячих малюнках, що не були використані при навчанні моделі (рис. 3.19).

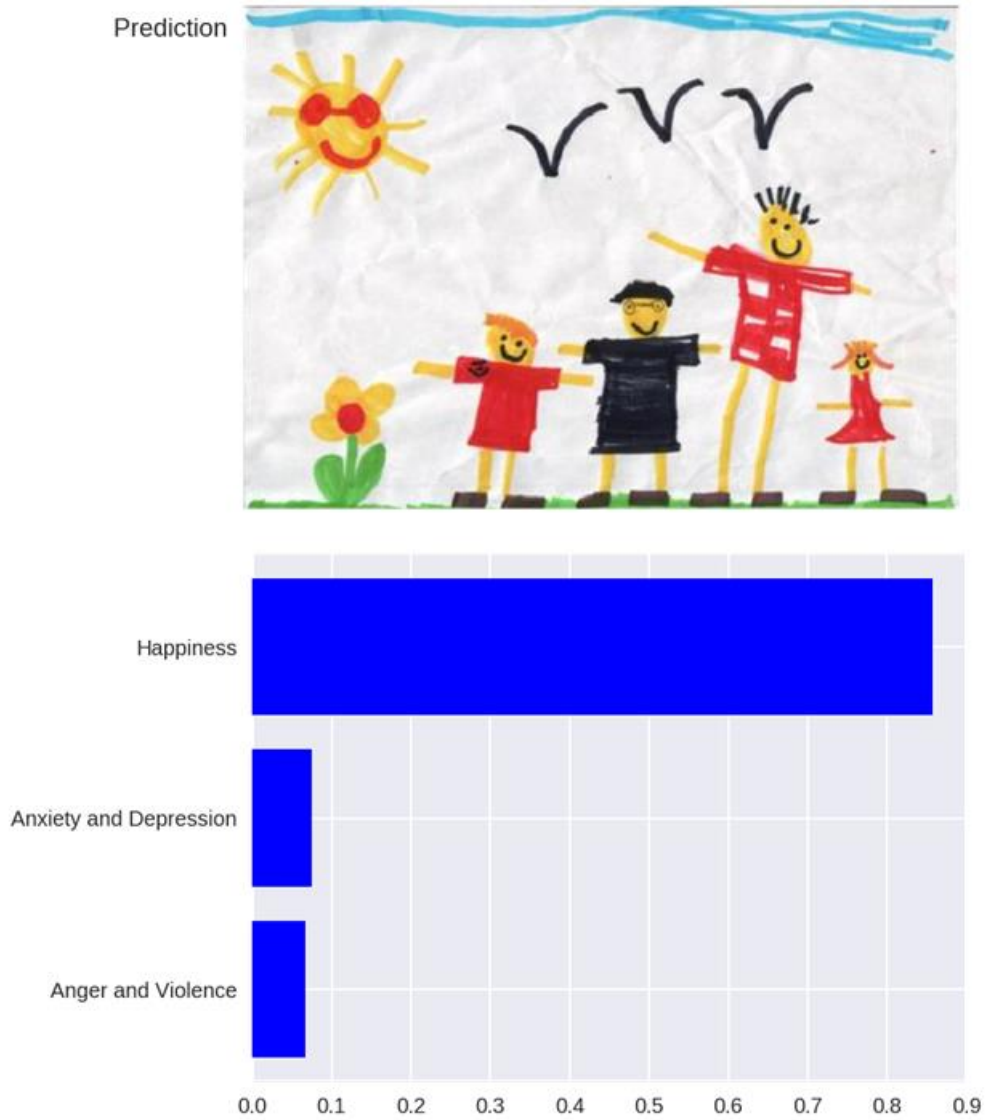


Рисунок 3.19 – Результат роботи моделі, категорія «Щастя».

Люди, що посміхаються, яскраві кольори, сонце – стовідсотковий класс «Щастя», який був розпізнаний нейронною мережею.

Розглянемо ще приклад, коли модель також розпізнала категорію «Щастя» (рис. 3.20).

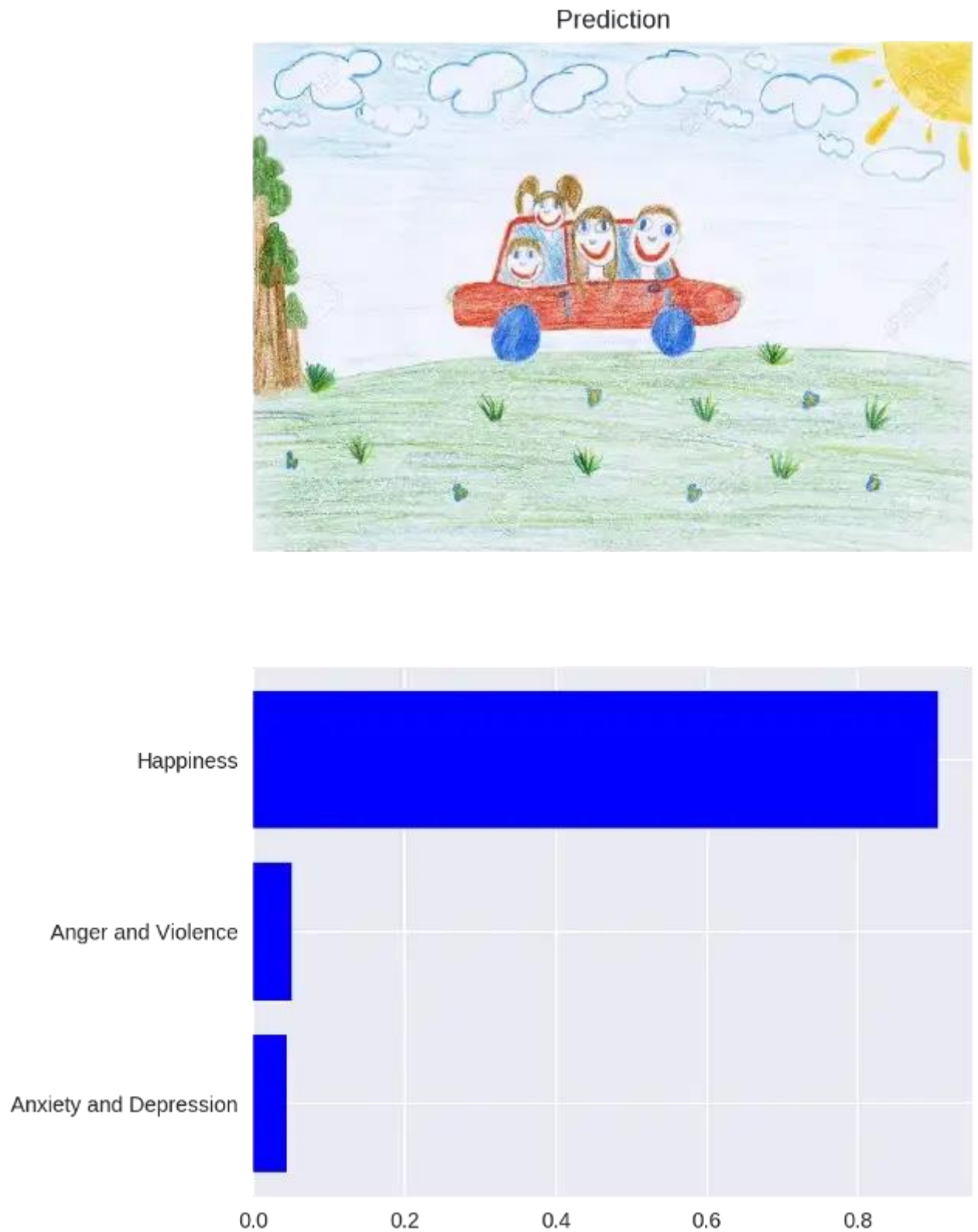


Рисунок 3.20 – Результат роботи моделі, категорія «Щастя».

Аналогічна ситуація з попереднім прикладом.

Наступними було спробовано віддати на вхід мережі малюнки, які очікується віднести до класу «Гнів і насильство» (рис. 3.21).

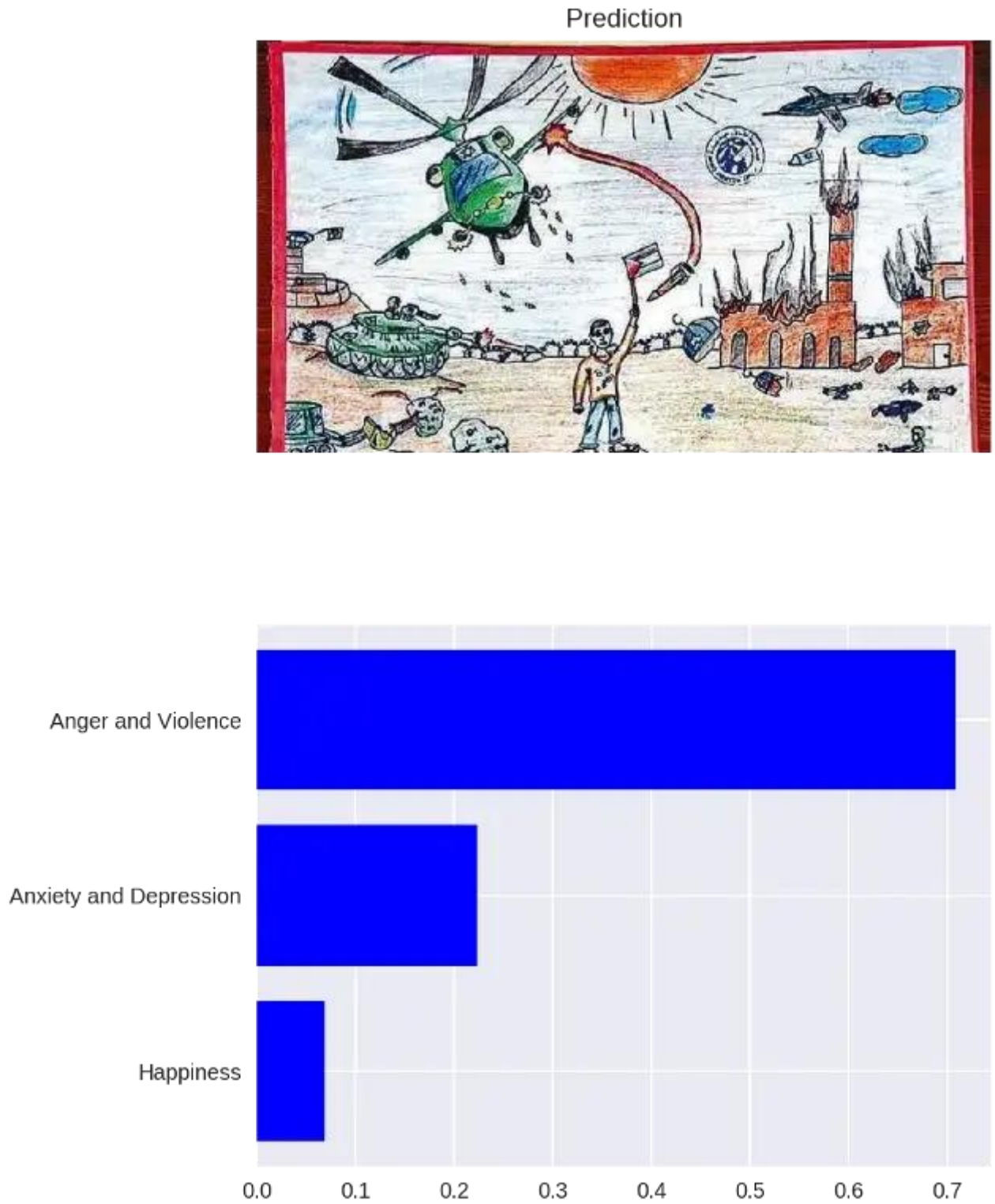


Рисунок 3.21 – Результат роботи моделі, категорія «Гнів і насильство».

Зображення війни, модель впоралась і правильно класифікувала зображення.

Ще приклад цієї категорії на рисунку 3.22:

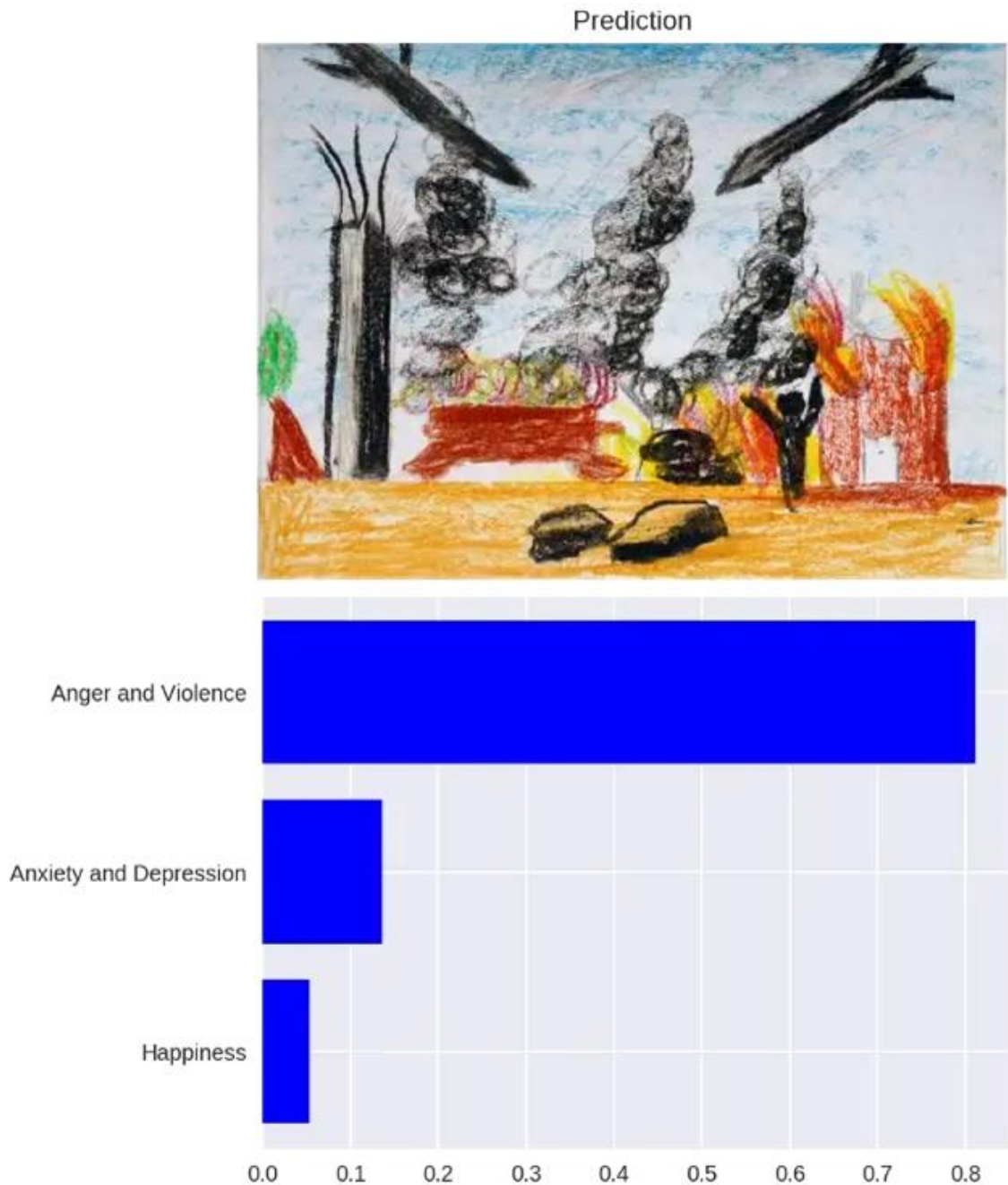


Рисунок 3.22 – Результат роботи моделі, категорія «Гнів і насильство».

Розглянемо кілька прикладів останнього класу «Тривога та депресія» (рис. 3.23).

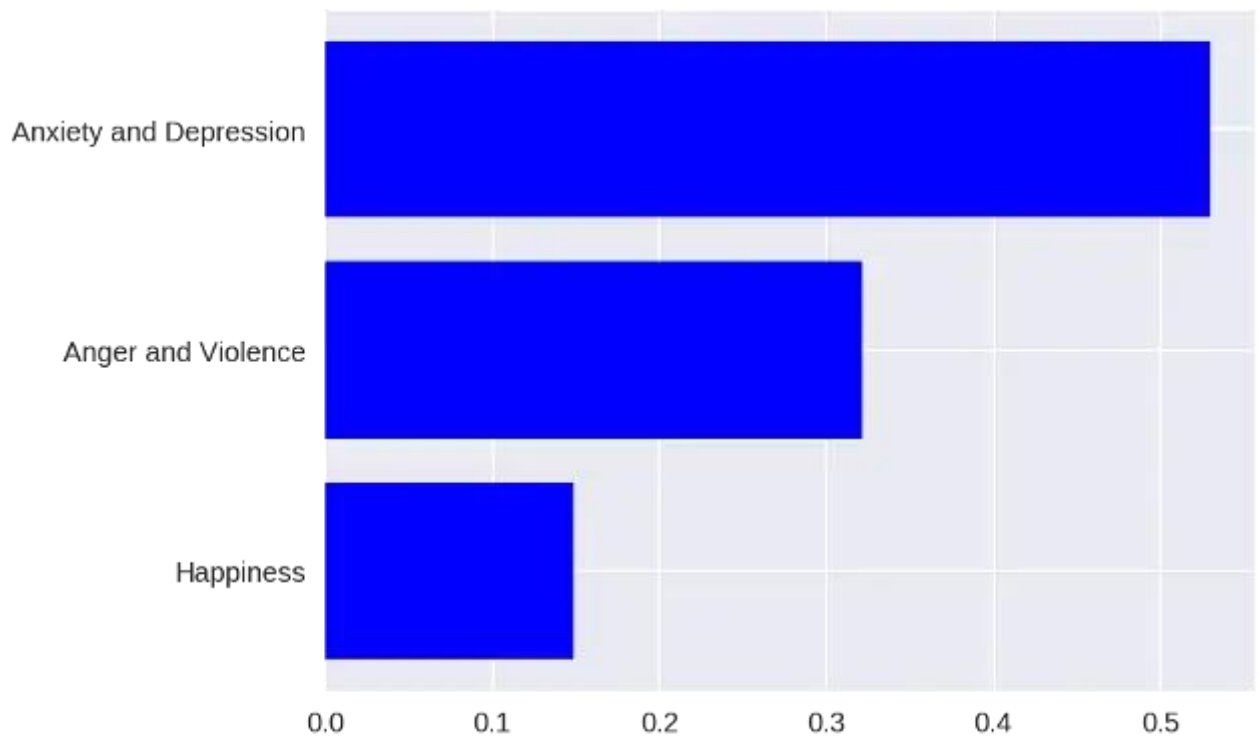
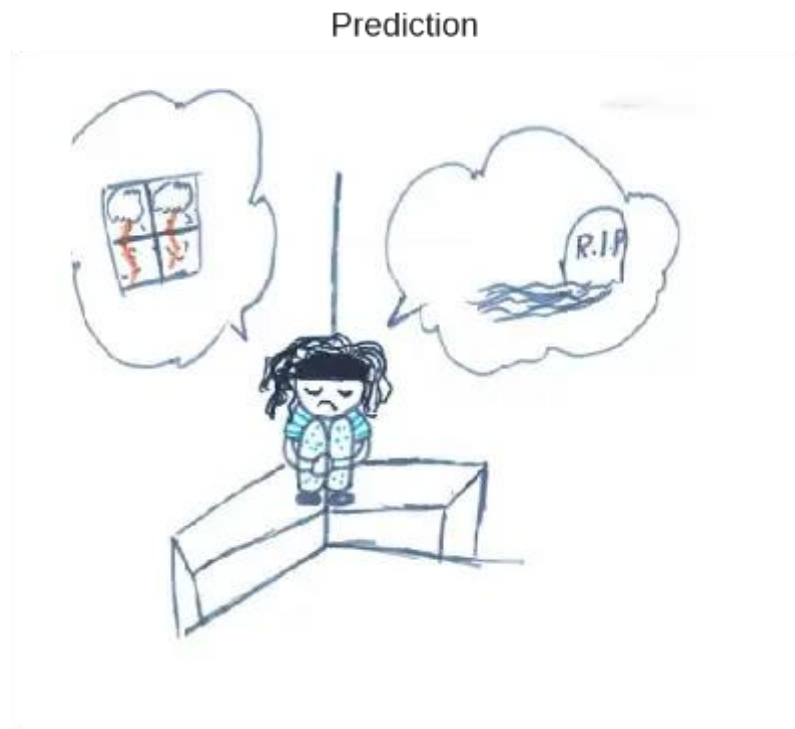


Рисунок 3.23 – Результат роботи моделі, категорія «Тривога та депресія».

Сумні обличчя, бідна кольорова палітра є характерною ознакою цього

класу. Також результат демонструє досить високу належність до класу «Гнів і насильство», проте домінуючий клас все одно був обран правильно.

Ще один приклад на рисунку 3.24:

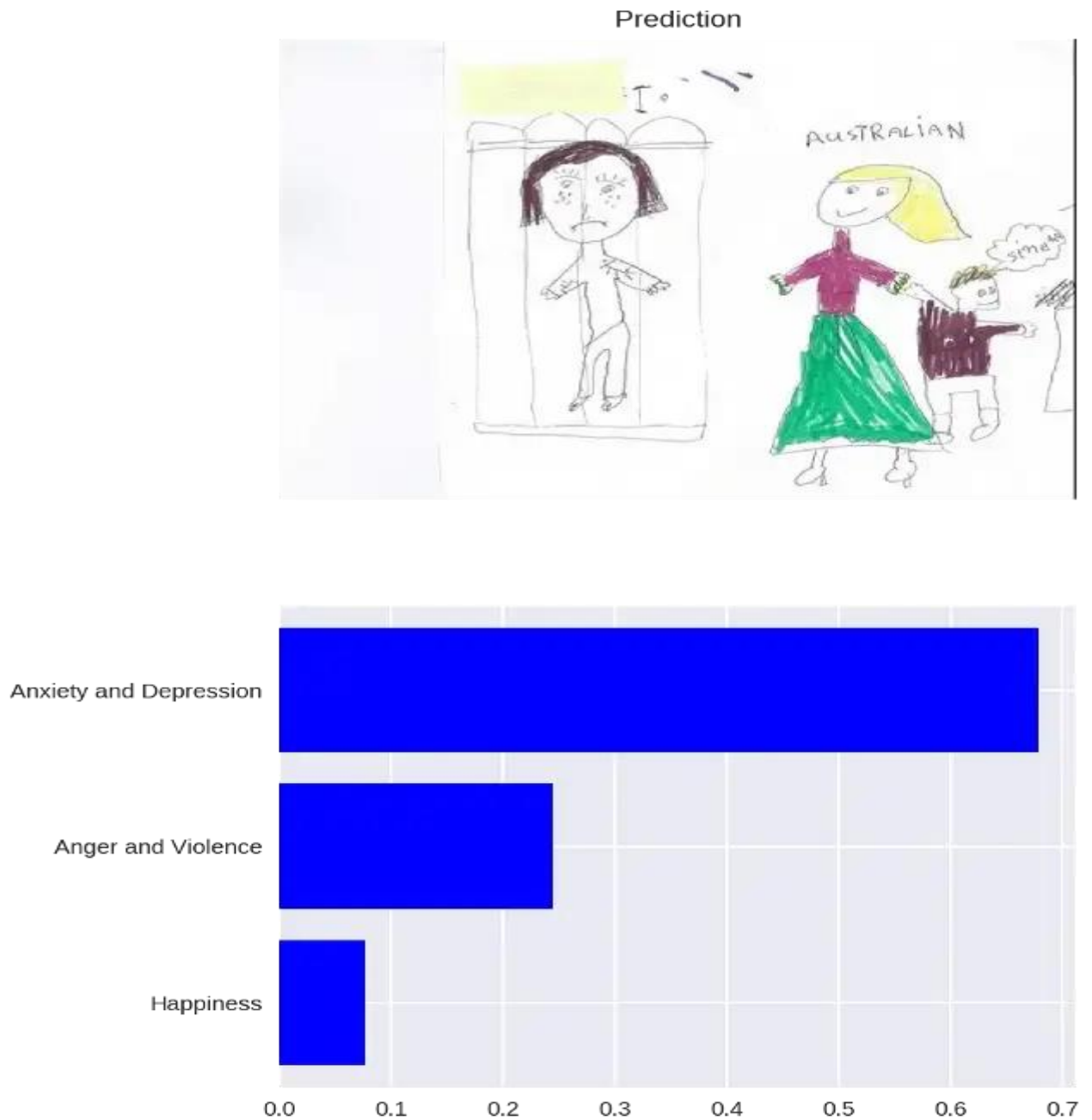


Рисунок 3.24 – Результат роботи моделі, категорія «Тривога та депресія».

Так само дуже характерним є наявність сумного обличчя на малюнку, тонкі лінії, неяскові кольори.

Отже в цьому розділі було розглянуто результати роботи створеної нейронної мережі, яка навчалась на спеціально відібраному датасеті. За результатами, що продемонстровано, можна сказати що дійсно модель досить чітко розподіляє дитячі малюнки на відповідні класи.

ВИСНОВКИ

У ході цієї роботи була розроблена нейромережева модель аналізу дитячих малюнків для діагностики емоційного стану дитини. У роботі була проведена дослідницька робота з огляду існуючих нейронних мереж, знайдено тип нейронної мережі, який найбільше підходить для вирішення завдання роботи (згортова нейронна мережа). Також в ході роботи було доведено її актуальність та наведено практичні ситуації, в яких створена мережа дійсно може принести користь для її користувачів.

Інший досліджений аспект, що був потрібний для створення такої нейромережевої моделі, це характерні ознаки, за якими фахівці власноруч детермінують емоційний стан дитини. Після дослідження цих ознак були відібрані малюнки, що є характерними для здорового стану та з певними відхиленнями; на їх основі створено датасет і навчено модель.

Враховуючи усе вищеперераховане, на основі цієї інформації, було дійсно розроблено, навчено і протестовано нейромережеву модель. В останньому розділі роботи наведено приклади роботи мережі, що доводить високу ймовірність отримання правильного результату на виході з мережі.

Звичайно, розроблена модель не може замінити фахівців. Питання емоційного здоров'я дітей є набагато глибшим аніж його діагностика за допомогою аналізу малюнків. Це величезне питання, яке роками досліджується різними фахівцями, спеціалістами та відповідними професорами. Але, як було зазначено в ході роботи, розроблена нейромережева модель аналізу дитячих малюнків дійсно може допомогти провести первинну діагностику дитини, допоможе батькам звернути увагу на це питання та вчасно звернутися до фахівців.

Варто відзначити, що наврядчи існують стовідсотково бездоганні системи та нейромережеві моделі, отже, завжди є аспекти, які можна вдосконалювати. Зазвичай, існує дві категорії вдосконалень або оновлень –

вертикальні та горизонтальні.

Вертикальними вдосконаленнями називають поліпшення самої системи, або продукту, або, в нашому випадку, нейромережевої моделі, яка була створена в рамках цього дипломного проекту. До вертикальних майбутніх імовірних поліпшень можна додати:

- збільшення даних (малюнків) у датасеті на якому навчається нейронна мережа. Дійсно, цей фактор ледь не найбільше впливає на кінцеву правильність отриманого результату в ході роботи моделі. Але потрібний нам датасет є досить специфічним, як зазначалося раніше, пошук необхідних дитячих малюнків, особливо для класів «Тривога і депресія» та «Гнів і насильство», не є легким завданням;

- розподіл за статтю та віком. На рисунки, звичайно, впливає стать дитини, адже навіть з проаналізованих робіт фахівців можна побачити, що вони дійшли до висновків, що, наприклад, кольори та форми фігур відрізняються у хлопців та дівчат, що є цілком логічним явищем. Так само вік: в статтях спеціалістів вказані переважаючи теми малюнків дітей різного віку, тож досить неоднозначні рисунки можуть просто бути ознакою певного періоду становлення дитини цього віку, що не означає наявності яких-небудь проблем у дитини.

Горизонтальними вдосконаленнями називають додавання інших систем, які будуть працювати разом з вже створеною. Такими ймовірними поліпшеннями можуть стати, наприклад, моделі, які будуть визначати емоційний стан дитини за виразом обличчя, рухом, мовленням тощо.

У ході виконання кваліфікаційної роботи експериментально доведено, що побудована нейромережева модель аналізу дитячих малюнків для діагностики емоційного стану дитини є працездатною і має потенціал розвитку, а саме впровадження вертикальних та горизонтальних вдосконалень для поліпшення її роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дашенкова Н. М. Психологія: навч. пос. – Харків: ХНУРЕ, 2008. – 172 с.
2. Кулініч І. О. Психологія управління: навч. посіб. – Київ: Знання, 2008. – 292 с.
3. Fawcett T. An introduction to ROC analysis: Pattern Recognition Letters / USA, 2006, pages 861-874.
4. Xiao-Yuan Jing, Fei Wu, Dong Xiwei, Baowen Xu. An Improved SDA Based Defect Prediction Framework for Both Within-Project and Cross-Project Class Imbalance Problems. IEEE Transactions on Software Engineering. 2016.
5. Субботін С. О. Нейронні мережі: теорія та практика: навч. посіб. – Житомир: Вид. О. О. Євенок, 2020. – 160 с.
6. CS231n: Convolutional Neural Networks for Visual Recognition [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу : <http://cs231n.github.io/neural-networks-1/>
7. Daily SB, James MT, Cherry D, Porter JJ, Darnell SS, Isaac J, et al. Chapter 9-affective computing: historical foundations, current applications, and future trends. Emotions and Affect in Human Factors and Human-Computer Interaction. Cambridge, MA: Academic Press (2017). p. 213–310.;
8. S. B. Kibria and M. S. Hasan, “An analysis of feature extraction and classification algorithms for dangerous object detection,” in Proceedings of the 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE), pp. 1–4, IEEE, Rajshahi, Bangladesh, December 2017.
9. H. O. Siyaka, O. Owolabi and H. I. Bisallah, ”A Novel Facial Image Deviation Estimation and Image Selection Algorithm (Fide-Isa) for Training Data Reduction in Facial Recognition System”; Journal of Science Technology and Education ISSN: 2277-0011; 9 (2), June, 2021.
10. Нильсен М.А. Нейронні мережі та глибоке навчання. - Determination

Press. -2015.

11. Лесна Н. С., Репка В. Б., Шатовська Т. Б. Інтелектуальний аналіз даних: навч. пос. – Харків: ХНУРЕ, 2003. – 112 с.

12. O. C. Oguine, K. J. Oguine, H. I. Bisallah and D. Ofuani, “Hybrid Facial Expression Recognition (FER2013) Model for Real-Time Emotion Classification and Prediction”, arXiv.org, 2022. [Online].

13. G. K. Verma and A. Dhillon, “A handheld gun detection using faster RCNN deep learning,” in Proceedings of the 7th International Conference on Computer and Communication Technology, pp. 84–88, Kurukshetra, Haryana, November 2017.

14. Корабльов М.М. Інтелектуальна система підтримки прийняття клінічних рішень на основі мультиагентного підходу та міркувань по прецедентам // Сучасні інформаційні технології і системи: монографія / за заг. ред. В.С. Пономаренка. – Х.: ХНЕУ ім. С. Кузнеця, 2022. – С. 139-164.

15. Mykola Korablyov, Natalia Axak, Oleksandr Fomichov and Andrii Chuprina. Hybrid Neuro-Fuzzy Model with Immune Training for Recognition of Objects in an Image / Proceedings of the 9th International Conference "Information Control Systems & Technologies", Odessa, Ukraine, September 24–26, 2020. – pp. 267-281

16. Кораблев Н.М., Фомичев А.А., Соловьев Д.Н., Чуприна А.А. Гибридные модели принятия решений с использованием иммунного подхода // Информационные управляющие системы и технологии. Проблемы и решения: монография. Под науч. ред. проф. Вычужанина Владимира. – Одесса: Экология, 2019. – С. 100-116.

17. Korablyov, M., Axak, N., Soloviov, D. Hybrid evolutionary decision-making model based on neural network and immune approaches (2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 1,8526594, с. 378-381.