

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка системи автоматичної корекції зображень на основі штучних нейронних мереж.
(тема)

Виконав:
студент 2 курсу, групи СШМ-18-3
Сосонний К.С.
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо - наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва спеціалізації)

Керівник проф. Терзіян В.Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Штучного інтелекту _____
(повна назва)

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 – Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо – наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системи штучного інтелекту (СШІ) _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Сосонного Костянтина Сергійовича _____
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи автоматичної корекції зображень на основі штучних нейронних мереж

затверджена наказом університету від 30 березня 2020 р. № 480 СТ

2. Термін подання студентом роботи до екзаменаційної комісії _____ 2020 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо дослідження та реалізації методів розпізнавання образів

4. Перелік питань, що потрібно опрацювати в роботі Аналіз предметної області та побудова технічного завдання, розробка структурної схеми системи, вибір технології та засобів розробки, розробка програмних модулів, розробка рекомендацій щодо підбору параметрів роботи алгоритму.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____

Рисунок 3.1 – Результат виконання програми

6. Консультанти розділів роботи (п.6 включається до завдання 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
Основна частина	проф. Терзіян В.Я.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	30.03.20	Виконано
2	Аналіз предметної області	30.03.20-09.04.20	Виконано
3	Постановка завдання	09.04.20-15.04.20	Виконано
4	Дослідження методів розпізнавання образів	25.05.20-19.05.20	Виконано
5	Розробка методу розпізнавання образів	19.05.20-27.05.20	Виконано
6	Розробка програмного забезпечення	27.04.20-04.05.20	Виконано
7	Написання пояснювальної записки	04.05.20-13.05.20	Виконано
8	Нормоконтроль	14.05.20	Виконано
9	Попередній захист	19.05.20	Виконано
10	Захист перед ЕК	21.05.20	

Дата видачі завдання 30 березня 2020 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Терзіян В.Я.
(посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 61 сторінок, 12 рисунків, 24 формул, 12 джерел, 2 додатки.

ГРАДІЄНТ ПОЛІТИКИ, РОЗПІЗНАВАННЯ ОБРАЗІВ, НЕЙРОННІ МЕРЕЖІ

В даному документі міститься опис методу обробки зображення. Проведено аналіз предметної області, в якому наведено огляд існуючих систем, дана оцінка застосовності існуючих систем. Розроблений метод розпізнавання образів для системи коректування зображень. Розроблено програмне забезпечення та проведено тестування розробленого програмного забезпечення. Розраховані економічні показники ефективності впровадження даного програмного продукту. Також розглянуті питання охорони праці, техніки безпеки і цивільної оборони.

РЕФЕРАТ

Пояснительная записка: 61 страниц, 12 рисунков, 24 формул, 12 источников, 2 приложения.

ГРАДИЕНТ ПОЛИТИКИ, РАСПОЗНАВАНИЕ ОБРАЗОВ, НЕЙРОННЫЕ СЕТИ.

В данном документе содержится описание метода обработки изображения. Проведен анализ предметной области, в котором приведен обзор существующих систем, дана оценка применимости существующих систем. Разработанный метод распознавания образов для системы корректировки изображений. Разработано программное обеспечение и проведено тестирование разработанного программного обеспечения. Рассчитаны экономические показатели эффективности внедрения данного программного продукта. Также рассмотрены вопросы охраны труда, техники безопасности и гражданской обороны.

ABSTRACT

Explanatory note: 61 pages, 12 picture, 24 Formula, 12 sources, 2 apps.

POLICY GRADIENT, PATTERN RECOGNITION, NEURAL NETWORKS

This document describes the image processing method. The analysis of the subject area, which provides an overview of existing systems, assesses the applicability of existing systems. Image recognition method for image correction system is developed. The software was developed and the software developed was tested. Economic indicators of efficiency of implementation of this software are calculated. The issues of labor protection, safety and civil defense were also considered.

ЗМІСТ

Перелік позначень і скорочень	9
Вступ	10
1. Аналіз предметної області	11
1.1 Якісний опис задачі розпізнавання образів	11
1.2 Сприйняття образів за сукупністю ознак	15
1.3 Біологічний нейрон і нейронні мережі	17
1.4 Аналогія між комп'ютером і людським мозком	20
1.5 Штучні нейронні мережі	22
1.6 Архітектура штучних нейронних мереж	27
2. Розробка методу розпізнавання образів для системи корегування зображень	31
2.1 Уявлення зображень	31
2.2 Редагування на рівні людини: Крива	36
2.3 Policy Gradient	37
2.4 Дискримінатор для змагального навчання	37
2.5 Навчання змагальності з Policy Gradient	37
2.6 Оптимізація	42
2.7 Інтуїція	45
3. Розробка системи корегування зображень	47
3.1 Особливості та переваги використання мови програмування Python	47
3.2 Python у порівнянні з іншими мовами	49
3.3 Розробка програми	50

	8
Висновки	52
Список джерел інформації	53
Додаток А	55
Додаток Б	60

ПЕРЕЛІК ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ЕОМ – Електронно обчислювальна машина;

РО – Розпізнавання образів;

ШНС – штучна нейронна мережа;

PG – Policy Gradient.

ВСТУП

Досить тривалий час завдання розпізнавання образів розглядалася тільки з біологічної точки зору. При цьому спостереженнями піддавалися лише якісні характеристики, які не дозволяють описати механізм функціонування.

Теорія розпізнавання образів є одним з основних розділів кібернетики як в теоретичному, так і в прикладному плані. Так, автоматизація деяких процесів передбачає створення пристроїв, здатних реагувати на мінливі характеристики зовнішнього середовища деякою кількістю позитивних реакцій.

Далі математична база теорії розпізнавання розширювалася за рахунок застосування розділів прикладної математики, теорії інформації, методів алгебри логіки, математичного програмування і системотехніки.

В умовах сьогодення проблема розв'язання задачі розпізнавання образів в системі корегування зображень є надзвичайно актуальною, особливо враховуючи постійну зайнятість населення та бажання ефективно використовувати свій час.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Якісний опис задачі розпізнавання образів

Образи – це об’єкти та явища навколишнього середовища або продукти розумової діяльності людини. Прикладом образів можуть бути: люди (їх голоси, почерки, пульс, хвороби, результати розумової діяльності, стилі малювання); літери; літаки; судна; автомашини; рослинність; виробничі, соціальні, суспільні, економічні явища та процеси; рукописи, висловлювання думок (у книжках).

Ознаки образу – це якісні та кількісні параметри, сукупність яких дозволяє відрізнити один образ від іншого, робить образ індивідуальним. Ознаки можуть бути суттєвими (вони найбільшою мірою визначають індивідуальність, неповторність образу) і несуттєвими (вони меншою мірою впливають на визначення, розпізнавання і класифікацію образу), кількісними, якісними, апріорними, апостеріорними.

Ознаки, які визначаються до випробування, зветься апріорними. Наприклад, до проведення іспиту системи розпізнавання образів (РО) з різних джерел (книг, креслень, випробувань і т.д.) отримуються апріорні ознаки літака (габарити, кількість і тип двигунів тощо).

Ознаки, які отримуються в результаті проведення випробування системи РО, зветься емпіричними, або апостеріорними, тобто отриманими за результатами дослідів. Наприклад, як результат проведення іспиту датчики системи РО видають апостеріорні ознаки конкретного літака (габарити, кількість і тип двигунів тощо). Для визначення класу образу система РО порівнює апостеріорні ознаки з апріорними ознаками.

Початковий словник ознак – це найбільш повний (у розумних межах, тобто забезпечений існуючими відповідними датчиками або реальною

можливістю їх створення та використання) перелік ознак образу, які отримуються за допомогою датчиків. У системах РО використовуються детерміновані, ймовірнісні, логічні, структурні та комбіновані ознаки.

Робочий словник ознак – це перелік ознак, які обрані з початкового словника ознак і використовуються у системі РО. Первинні ознаки образу, отримані первинними датчиками, у системі РО можуть бути об'єктом інтелектуальної обробки і переробляти у вторинні ознаки (об'єднуватись, роз'єднувати, фільтрувати), в результаті чого створюється реальний простір ознак, який, власне, і використовується системою РО.

Однотипні образи – це група образів, яка описується кінцевою кількістю лише їм властивих ознак (своїм початковим словником ознак) і об'єднаних приблизно однаковими функціями та ідеєю призначення. Приклади однотипних образів: літаки, автомашини, люди, птахи і т.д.

Клас образів – це підмножина образів з найбільш схожими суттєвими ознаками, обрана з множини однотипних образів за допомогою робочого словника ознак. Приклади класів образів: однотипні образи «Автомобілі» розділяються на класи «Волга», «Лада», «Запорожець» і т.д. Часто до існуючих m класів додається $(m+1)$ -й «нерозпізнаний образ». Це означає відмову від управління, що іноді краще, ніж помилкове визначення ситуації. Сукупність однотипних класів складають алфавіт класів.

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

Об'єктом дослідження РО є образи: об'єкти, предмети чи явища навколишнього середовища або предмети розумової діяльності людини.

Предметом дослідження курсу РО є методи та засоби РО.

Методи дослідження курсу РО засновані на теорії ймовірностей та математичної статистики; матричній алгебрі; диференціальному, інтегральному, операційному численні; методах роботи на ЕОМ.

Метою РО є розробка методів та засобів для точної класифікації образів з наступною задачею прийняття рішень по виконанню практичних дій, які були б адекватною реакцією на стан навколишнього середовища. Система РО є складовою частиною штучного інтелекту, який за своїми можливостями наближається до можливостей людини або будь-якої живої істоти, життя якої складається з безперервного виконання алгоритму: «виділити та класифікувати образ – прийняти рішення – діяти». Метою РО є автоматизація виконання операцій; підвищення якості та швидкості виконання роботи.

Типовими задачами РО є розпізнавання і розподіл на класи: літери; текст; літаки; автомашини; птахи; обличчя, голоси, почерки, стилі письма, хвороби людини; елементи складних пристроїв, що вийшли з ладу (діагностика пристроїв); елементи складних пристроїв, які у майбутньому почнуть ненормально працювати (прогнозування роботи пристроїв); виробничі, соціальні, суспільні, економічні явища та процеси і т.п.

Звичайно, найкращою системою РО є людина й інші живі істоти. Людина протягом всього свого життя розпізнає і класифікує за ознаками сигнали, об'єкти, явища, процеси, ситуації та інші образи. Людин може розпізнати (класифікувати) дерева, квіти, траву; розгадати шифр; прочитати друкований і рукописний пошкоджений або зіпсований текст (навіть з відсутніми літерами і словами та частково втраченими фразами); виконати медичну, технічну, економічну, соціальну діагностику і т.п.

Задачі РО розв'язують і інші живі істоти: звіри, риби, комахи. Навіть насіння та рослини «пізнають і класифікують» пори року: весну, літо, осінь, зиму. Їх образ життя у принципі не відрізняється від образу життя

живої істоти: «виділити та класифікувати образ – прийняти рішення – діяти».

Засоби, які використовуються при РО розподіляються на три групи:

- технічні датчики (зображення, температури, розміру, кольору, швидкості руху, тиску, ваги, магнітного поля і т.п.), які дозволяють виділити і «зважити» суттєві первинні ознаки образу;

- технічні засоби інтелектуальної обробки (переробки, порівняння, об'єднання, переоформлення, накопичення) інформації, в результаті чого отримують вторинні ознаки, у просторі яких здійснюється, власне, класифікація;

- математичне та програмне забезпечення у складі методів, вирішальних правил, обмежень, алгоритмів, програм, які використовуються для визначення класу образу.

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

Раніше вважалося, що абстрактне мислення, узагальнення, логіка, навчання на прикладах, інтелектуальна обробка даних властиві лише людині. Тепер ситуація змінилась, бо ЕОМ також може розв'язувати аналогічні проблеми.

Але треба визначити, що технічні засоби не мають такої універсальності, яка притаманна живій істоті: якщо технічна система РО класифікує літаки, то вона не може розпізнавати текст або голос людини. Тобто технічні засоби працюють лише за вузьким вказаним призначенням.

1.2 Сприйняття образів за сукупністю ознак

Під сприйняттям розуміють процес пізнання за допомогою чуттєвих органів або висновків. Очевидно, що ця операція є первинною у процедурі РО, бо класифікувати можна лише те, що вже усвідомлене як сукупність ознак, які належать до однотипних об'єктів або явищ.

Виділити об'єкт з навколишнього середовища можна за допомогою сприйняття й усвідомлення однієї ознаки або сукупності ознак, що йому належать. Людина сприймає ці ознаки за допомогою чуттєвих органів (зору, слуху, дотику, смаку, нюху) та розуму. Розум відіграє велику роль у розпізнаванні. Наприклад, з метою класифікації оцінку габаритів якогось об'єкта «на око» одна людина може вказати з похибкою відносно дійсних розмірів у 5%, а інша людина (навіть з кращим зором) – у 10-50%. Це пояснюється набутим досвідом, розумом. Справа ускладнюється, якщо множина об'єктів, які належить класифікувати (розбити на групи, класи), характеризується сукупністю ознак. Але й тут є люди, які за сукупністю ознак (по хмарах на небі, по вітру, по реагуванню рослин, комах, птахів) можуть передбачити (класифікувати) майбутню погоду. Лікарі за сукупністю ознак визначають (класифікують) хворобу людини. Майстер за сукупністю ознак приблизно знає, яке пошкодження треба шукати у телевізорі (класифікує пошкодження). Фінансист за сукупністю окремих ознак визнає (класифікує), що фінансове становище є загрозливим або не загрозливим.

Всьому цьому можна навчитись. Але основна проблема полягає у тому, щоб цьому навчити машину. Машина повинна сприймати процеси, образи й явища навколишнього середовища і робити висновки: класифікувати їх. Це є одна з ключових проблем штучного інтелекту.

Швидкість та ефективність класифікації образу залежить від того, наскільки добре підібрані суттєві ознаки на попередньому етапі.

Щоб розпізнавати образи, людина повинна виділити образ з навколишнього середовища, визначити і виміряти ознаки образу за допомогою «датчиків» і за ознаками класифікувати образ. Як датчики, які пов'язують людину з навколишнім середовищем, використовуються чуттєві органи:

- зору – для визначення розмірів, об'єму, кольорів, інтенсивності кольорів, нерухомості, швидкості і напрямку переміщення, контрастності, яскравості, структури;

- слуху – для визначення інтенсивності звуку, однотонність, тембр, музики, висоти звуку, шуму, безперервності, мовлення, співу;

- дотику – для визначення гладкості, липкості, грубозності, жорсткості, м'якості, твердості, шороховатості, пружності, тиску, сили, текстури, границі, легкості переміщення, розмірів, об'єму;

- смаку – для визначення приємності, неприємності, інтенсивності, специфічності, побочних ефектів;

- нюху – для визначення приємності, неприємності, інтенсивності, специфічності, побочних ефектів.

До «датчиків» можна віднести також розум людини, який переробляє, об'єднує, фільтрує інформацію за сукупністю всіх отриманих ознак.

Але розум людини розв'язує далі не менш складну і відповідальну задачу: на базі отриманої сукупності ознак він класифікує образи.

Системи РО є технічними об'єктами, які деякою мірою моделюють вказані властивості людини. Треба визнати, що у загальному плані за компактністю, терміном служби, надійністю та іншими показниками

технічні датчики є менш досконаліми у порівнянні з органами людини. Але за окремими показниками технічні датчики перевершують.

1.3 Біологічний нейрон і нейронні мережі

Кожен нейрон має багато якостей, спільними з іншими елементами тіла, але його унікальною здатністю є приймання, обробка і передача електрохімічних сигналів по нервових шляхах, які утворюють комунікаційну систему мозку. На рисунку 1.1 і рисунку 1.2 показана структура і будова типових біологічних нейронів. Нейрон складається з трьох частин: тіла клітини, дендритів і аксонів, кожна частина зі своїми, але взаємопов'язаними функціями. Дендрити йдуть від тіла нервової клітини (сома) до інших нейронів, де вони приймають сигнали в точках з'єднання, які називаються синапсами.

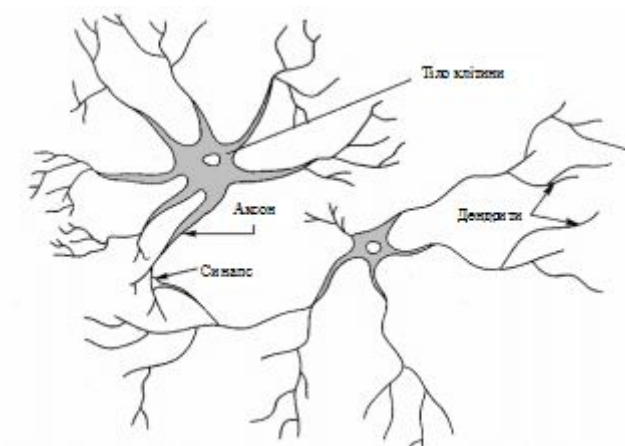


Рисунок 1.1 – Структура типових біологічних нейронів

Прийняті синапсом вхідні сигнали підводяться до тіла нейрона. Тут вони підсумовуються, причому одні входи прагнуть порушити нейрон, інші – перешкодити його порушення. Коли сумарне збудження в тілі

нейрона перевищує деякий поріг, нейрон збуджується, посилаючи по аксону сигнал іншим нейронам. У цієї основної функціональної схеми багато ускладнень і винятків, проте більшість штучних нейронних мереж моделюють лише прості властивості.



Рисунок 1.2 – Будова типових біологічних нейронів

Нервові клітини, або нейрони, являють собою особливий вид клітин в живих організмах, що володіють електричної активністю, основне призначення яких полягає в оперативному управлінні організмом. Сона, як правило, має поперечний розмір кілька десятків мікрон. Довжина дендритів може досягати 1 мм, дендрити сильно розгалужуються, пронизуючи порівняно великий простір в околиці нейрона. Довжина аксона може досягати сотень кілометрів. На соммі і на дендритах розташовуються закінчення (колатералі) аксонів, що йдуть від інших нервових клітин. Кожне таке закінчення має вигляд потовщення, званого синаптичної бляшкою, або синапсом. Поперечні розміри синапсу, як

правило, не перевищують декількох мікрон, найчастіше ці розміри становлять близько 1 мкм.

Вхідні сигнали дендритного дерева (постсинаптичні потенціали) зважуються і підсумовуються на шляху до аксонів горбок, де генерується вихідний імпульс (спайк) або пачка імпульсів. Його наявність (або інтенсивність), отже, є функцією зваженої суми вхідних сигналів. Вихідний сигнал проходить по гілках аксона і досягає синапсів, які з'єднують аксони з дендритними деревами інших нейронів. Через синапси сигнал трансформується в новий вхідний сигнал для суміжних нейронів. Цей вхідний сигнал може бути позитивним і негативним (збудливим або гальмують) залежно від виду синапсів. Величина вхідного сигналу, що генерується синапсом, може бути різною навіть при однаковій величині сигналу, що приходить в синапс. Ці відмінності визначаються ефективністю або вагою синапсу. Синаптична вага може змінюватися в процесі функціонування синапсу.

Нейрони можна розбити на три великі групи: рецепторні, проміжні і ефекторні. Рецепторні нейрони забезпечують введення в мозок сенсорної інформації. Вони трансформують сигнали, що надходять на органи чуття (оптичні сигнали в сітківці ока, акустичні в вушної равику або нюхові в хеморецепторах носа), в електричну імпульсація своїх аксонів. Ефекторні нейрони передають що приходять на них сигнали виконавчим органам. На кінці їх аксонів є спеціальні синаптичні з'єднання з виконавчими органами, наприклад м'язами, де збудження нейронів трансформується в скорочення м'язів. Проміжні нейрони здійснюють обробку інформації, одержуваної від рецепторів, і формують керуючі сигнали для ефекторів. Вони утворюють центральну нервову систему.

1.4 Аналогія між комп'ютером і людським мозком

Основні положення теорії діяльності головного мозку і математична модель нейрона були розроблені У. Мак-Каллоком і Ч. Піттсом в 1943 році. Згідно запропонованої моделі мозок являє собою безліч нейронів, що мають однакову структуру. Кожен нейрон реалізує деяку функцію, звану пороговою, над вхідними значеннями. Якщо значення функції перевищує певну величину – поріг (що характеризує сумарну значущість отриманої нейроном інформації), нейрон збуджується і формує вихідний сигнал для передачі його іншим нейронам. Пройшовши шлях від рецепторів (слухових, зорових і інших) через нейронні структури мозку до виконавчих органів, вхідна інформація перетворюється в набір керуючих впливів, адекватних ситуації.

Окремі нейрони, з'єднуючись між собою, утворюють нову якість, яке, в залежності від характеру міжнейронних з'єднань, має різні рівні біологічного моделювання:

- група нейронів;
- нейронна мережа;
- нервова система;
- розумова діяльність;
- мозок.

Існує подібність між мозком і цифровим комп'ютером: обидва оперують електронними сигналами, обидва складаються з великої кількості простих елементів, обидва виконують функції, які є, грубо кажучи, обчислювальними, тим не менше, існують і фундаментальні відмінності. У порівнянні з мікросекундними і навіть наносекундною інтервалами обчислень сучасних комп'ютерів нервові імпульси є занадто повільними. Хоча кожен нейрон вимагає наявності мілісекундного

інтервалу між переданими сигналами, висока швидкість обчислень мозку забезпечується величезним числом паралельних обчислювальних блоків, причому кількість їх набагато перевищує доступне сучасним ЕОМ. Діапазон помилок представляє інша фундаментальна відмінність: ЕОМ притаманна свобода від помилок, якщо вхідні сигнали бездоганно точні і її апаратне і програмне забезпечення не пошкоджені. Мозок же часто справляє краще вгадування і наближення при частково незавершених і неточних вхідних сигналах. Часто він помиляється, але величина помилки повинна гарантувати наше виживання протягом мільйонів років. Ці дві системи явно різняться в кожній своїй частині. Вони оптимізовані для вирішення різних типів проблем, мають істотні відмінності в структурі і їх робота оцінюється різними критеріями.

Розвиток штучних нейронних мереж надихається біологією. Структура штучних нейронних мереж була змодельована як результат вивчення людського мозку. Штучні нейронні мережі надзвичайно різноманітні за своїми формами, функціональності і цільовим призначенням. Розглядаючи мережеві конфігурації і алгоритми, дослідники мислять їх в термінах організації мозкової діяльності, але на цьому аналогія може і закінчитися. Подібність між ними дуже незначно, однак, навіть ця скромна емуляція мозку дає відчутні результати.

Наприклад, штучні нейронні мережі мають такі аналогічні мозку властивості, як здатність навчатися на досвіді, заснованому на знаннях, робити абстрактні умовиводи і робити помилки, що є більш характерним для людської думки, ніж для створених людиною комп'ютерів. Розробникам мереж доводиться виходити за межі сучасних біологічних знань в пошуках структур, здатних виконувати корисні функції. У багатьох випадках це призводить до необхідності відмови від біологічної правдоподібності, мозок стає просто метафорою, і створюються мережі,

неможливі в живій матерії або можуть бути використані неправдоподібно великих припущень про анатомію і функціонуванні мозку.

В даний час виникли і залишаються дві взаємно збагачують один одного мети нейронного моделювання: перша – зрозуміти функціонування нервової системи людини на рівні фізіології і психології і друга – створити обчислювальні системи (штучні нейронні мережі), що виконують функції, подібні до функцій мозку. Нейрони можна моделювати досить простими автоматами, а вся складність мозку, гнучкість його функціонування і інші найважливіші якості визначаються зв'язками між нейронами.

Кожна зв'язок представляється як простий елемент, службовець для передачі сигналу і його лінійного посилення або ослаблення.

Існує великий клас задач: нейронні системи асоціативної пам'яті, статистичної обробки, фільтрації та інші, для яких зв'язку формуються за певними формулами, при цьому навчання нейронних мереж виявилось здійсненним при моделюванні задач на звичайних персональних комп'ютерах.

1.5 Штучні нейронні мережі

Сьогодні, як і сто років тому, без сумніву, що мозок працює більш ефективно і принципово іншим чином, ніж будь-яка обчислювальна машина, створена людиною. Саме цей факт протягом стількох років спонукає і направляє роботи вчених усього світу зі створення і дослідження штучних нейронних мереж (ШНС).

До перших спроб розкрити секрет високої ефективності мозку можна віднести роботу Рамон-і-Кахаля (1911), в якій була висловлена ідея про нейрон як структурної одиниці мозку. Однак нейрон має на 5-6 порядків меншу швидкість спрацьовування, ніж напівпровідниковий логічний

елемент. Як показали пізніші дослідження, секрет високої продуктивності мозку полягає у величезній кількості нейронів і масивних взаємозв'язках між ними.

Мережа нейронів, утворює людський мозок, є високоефективною, комплексною, нелінійною, істотно паралельною системою обробки інформації. Вона здатна організувати свої нейрони таким чином, щоб реалізувати сприйняття образу, його розпізнавання або управління рухом, у багато разів швидше, ніж ці завдання будуть вирішені найсучаснішими комп'ютерами.

ШНС є спрощеною моделлю мозку. Вона будується на основі штучних нейронів, які володіють тим же основним властивістю, що і живі: пластичністю. Використання структури мозку і пластичності нейронів робить ШНС універсальною системою обробки інформації. У загальному випадку ШНС – це машина, що моделює спосіб роботи мозку. Зазвичай ШНС реалізуються у вигляді електронних пристроїв чи комп'ютерних програм. Серед багатьох можна виділити визначення ШНС як адаптивної машини, дане в: штучна нейронна мережа - це істотно паралельно розподілений процесор, який володіє природною схильністю до збереження досвідченого знання і можливістю надання його нам. Вона схожа з мозком в двох аспектах: знання здобувається мережею в процесі навчання, для збереження знання використовуються сили міжнейронних з'єднань,

Процедура, яка використовується для здійснення процесу навчання, називається алгоритмом навчання. Її функція полягає в модифікації синаптичних ваг ШНС певним чином так, щоб вона набула необхідних властивостей.

Модифікація ваг є традиційним способом навчання ШНС. Такий підхід близький до теорії адаптивних лінійних фільтрів, які вже давно і

успішно застосовуються в управлінні. Однак для ШНС існує ще і можливість модифікації власної топології, яка ґрунтується на тому факті, що в живому мозку нейрони можуть з'являтися, вмирати і змінювати свої зв'язки з іншими нейронами.

Зі сказаного вище стає ясно, що ШНС реалізують свою обчислювальну потужність, завдяки двом основним своїм властивостям: істотно паралельно розподіленій структурі і здатності навчатися і узагальнювати отримані знання. Під властивістю узагальнення розуміється здатність ШНС генерувати правильні виходи для вхідних сигналів, які не були враховані в процесі навчання (тренування). Ці дві властивості роблять ШНС системою переробки інформації, яка вирішує складні багатовимірні завдання, непосильні іншим технікам.

З точки зору машинного навчання, нейронна мережа являє собою окремий випадок методів розпізнавання образів, дискримінантного аналізу, методів кластеризації і т.п. З математичної точки зору, навчання нейронних мереж – це багатопараметрична завдання нелінійної оптимізації. З точки зору кібернетики, нейронна мережа використовується в задачах адаптивного управління і як алгоритми для робототехніки. З точки зору розвитку обчислювальної техніки та програмування, нейронна мережа – спосіб вирішення проблеми ефективного паралелізму. А з точки зору штучного інтелекту, ШНС є основою філософської течії конективізму і основним напрямком в структурному підході з вивчення можливості побудови (моделювання) природного інтелекту за допомогою комп'ютерних алгоритмів.

Нейронні мережі не програмуються в звичайному сенсі цього слова, вони навчаються. Можливість навчання – одне з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. В процесі

навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що, в разі успішного навчання, мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних і / або «зашумлених», частково перекручених даних.

Штучна нейронна мережа (ШНС, нейронна мережа) – це набір нейронів, з'єднаних між собою. Як правило, передавальні функції всіх нейронів у нейронній мережі фіксовані, а ваги є параметрами нейронної мережі і можуть змінюватися. Деякі входи нейронів позначені як зовнішні входи нейронної мережі, а деякі виходи – як зовнішні виходи нейронної мережі. Подаючи будь-які числа на входи нейронної мережі, ми отримуємо якийсь набір чисел на виходах нейронної мережі. Таким чином, робота нейронної мережі полягає в перетворенні вхідного вектора у вихідний вектор, причому це перетворення задається вагами нейронної мережі.

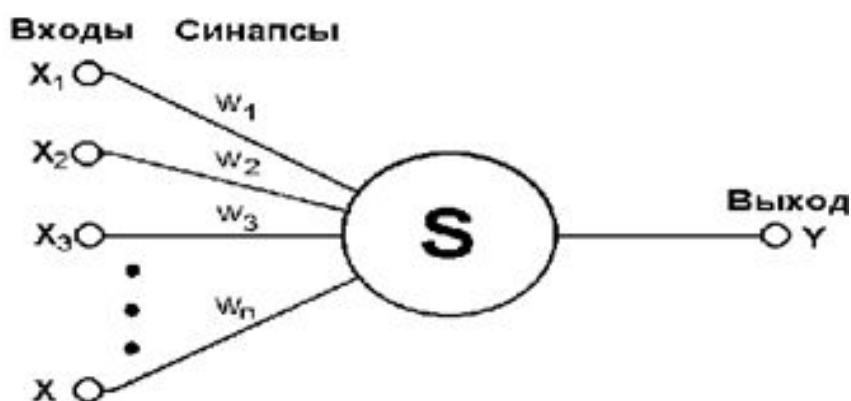


Рисунок 1.3 – штучний нейрон

Штучна нейронна мережа це сукупність нейронних елементів та зв'язків між ними (рисунок 1.3).

Основу кожної штучної нейронної мережі складають відносно прості, в більшості випадків – однотипні, елементи (комірки), що імітують роботу нейронів мозку (далі під нейроном ми будемо мати на увазі штучний нейрон, осередок штучної нейронної мережі).

Нейрон має групу синапсів – односпрямованих вхідних зв'язків, з'єднаних з виходами інших нейронів. Кожен синапс характеризується величиною синаптичної зв'язку або її вагою.

Кожен нейрон має поточний стан, яке зазвичай визначається, як зважена сума його входів:

$$s = \sum_{i=1}^n x_i * w_i \quad (1.1)$$

Нейрон має аксон – вихідну зв'язок даного нейрона, з якої сигнал (збудження або гальмування) надходить на синапси наступних нейронів. Вихід нейрона є функція його стану. Функція f називається функцією активації (рисунок 1.4).

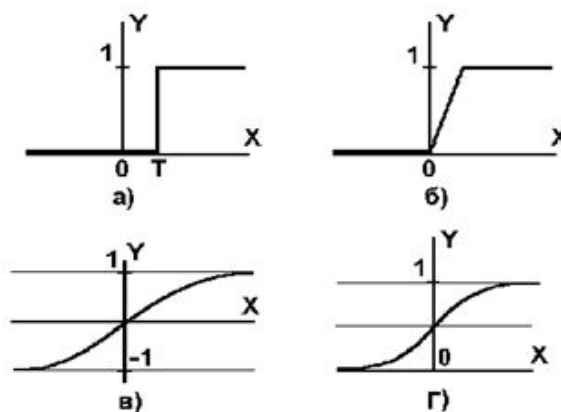


Рисунок 1.4 – функція активації

Функція активації може мати різний вигляд: пороговий (рисунок 1.4а), кусочно-лінійний (рисунок 1.4б), пігмоїд (рисунок 1.4в, 1.4г).

Безліч всіх нейронів штучної нейронної мережі можна розділити на підмножини – т.зв. шари. Взаємодія нейронів відбувається пошарово.

Шар штучної нейронної мережі – це безліч нейронів, на які в кожен такт часу паралельно надходять сигнали від інших нейронів даної мережі

Вибір архітектури штучної нейронної мережі визначається завданням. Для деяких класів задач вже існують оптимальні конфігурації. Якщо ж завдання не може бути зведена ні до одного з відомих класів, розробнику доводиться вирішувати задачу синтезу нової конфігурації. Проблема синтезу штучної нейронної мережі сильно залежить від завдання, дати загальні докладні рекомендації важко. У більшості випадків оптимальний варіант штучної нейронної мережі виходить дослідним шляхом.

Штучні нейронні мережі можуть бути програмного і апаратного виконання. Реалізація апаратна зазвичай являє собою паралельний обчислювач, що складається з безлічі простих процесорів.

1.6 Архітектура штучних нейронних мереж

ШНС може розглядатися як спрямований граф зі зваженими зв'язками, в якому штучні нейрони є вузлами. За архітектурою зв'язків ШНС можуть бути згруповані в два класи (рисунок 1.5): мережі прямого поширення, в яких графи не мають петель, і рекурентні мережі, або мережі з зворотними зв'язками.

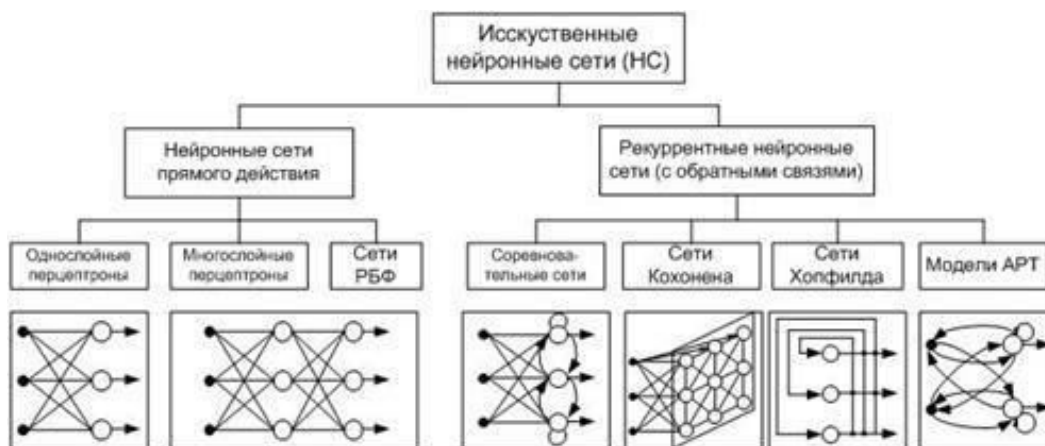


Рисунок 1.5 – архітектура зв'язків ШНС

У найбільш поширеному сімействі мереж першого класу, званих багат шаровим персептроном, нейрони розташовані шарами і мають односпрямовані зв'язку між шарами. На рисунку 1.5 представлені типові мережі кожного класу. Мережі прямого поширення є статичними в тому сенсі, що на заданий вхід вони виробляють одну сукупність вихідних значень, що не залежать від попереднього стану мережі. Рекурентні мережі є динамічними, так як в силу зворотних зв'язків в них модифікуються входи нейронів, що призводить до зміни стану мережі.

У загальному випадку поняття «штучна нейронна мережа» охоплює ансамблі нейронів будь-якої структури, однак практичне застосування знайшли лише деякі з них. Це пояснюється тим, що архітектура ШНС безпосередньо пов'язана з методом її навчання. Навіть різні етапи розвитку ШНС визначалися появою нових архітектур мереж і спеціально розроблених для них методів навчання.

Штучний нейрон (або просто нейрон) є елементарним функціональним модулем, з безлічі яких будуються ШНС. Він являє собою модель живого нейрона, однак лише в сенсі здійснюваних ним

перетворень, а не способу функціонування. Існують логічні, безперервні і імпульсні моделі нейрона. Логічні моделі нейрона (зокрема, описуваний картою Відень формальний нейрон) активно досліджувалась в 60-70-х роках, але не отримали подальшого розвитку. Імпульсні моделі ближчі до фізичну природу процесів, що відбуваються в нервовій клітині, проте їх теорія не так розвинена як у безперервних, і вони все ще не знаходять широкого застосування.

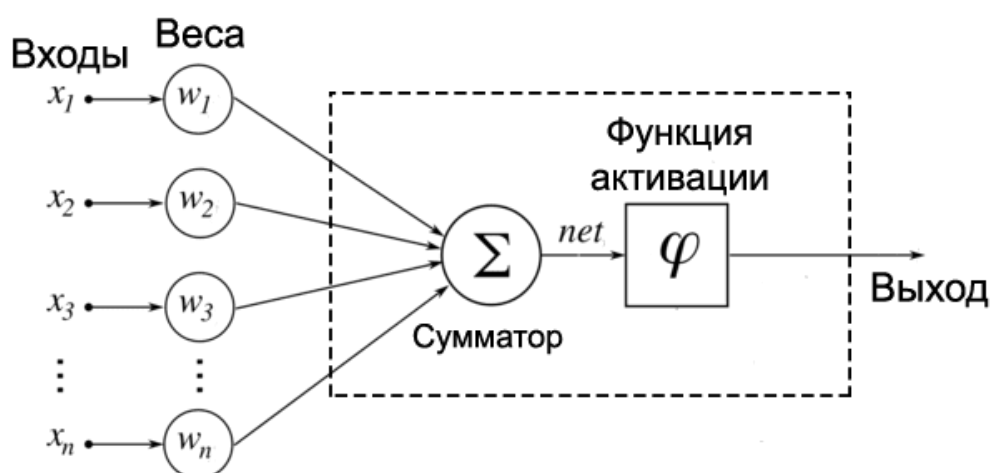


Рисунок 1.6 – Формальний нейрон

Адаптивний суматор обчислює скалярний добуток вектора вхідного сигналу на вектор параметрів. Адаптивним він називається через наявність вектора параметрів, що настроюються. Для багатьох задач корисно мати лінійну неоднорідну функцію вихідних сигналів, для цього додають постійний одиничний вхідний сигнал.

Нелінійний перетворювач сигналу – отримує скалярний вхідний сигнал і відповідним чином його перетворює.

Точка розгалуження служить для розсилки одного сигналу за кількома адресами. Вона отримує скалярний вхідний сигнал і передає його всім своїм виходів.

Формальний нейрон реалізує передавальну функцію:

$$\varphi(x) = \frac{1}{1+e^{-x}} \quad N(x_1, x_2, \dots, x_n) = \varphi \left(\sum_{i=1}^n \alpha_i * x_i + \alpha_0 \right) \quad (1.2)$$

З формальних нейронів можна скласти шари нейронів, які в свою чергу можна об'єднувати в багатошарові мережі (рисунок 1.7). Нейрони вхідного шару отримують сигнали, перетворюють їх і через точки розгалуження передають нейронам прихованого шару. Далі спрацьовує наступний шар аж до вихідного, який видає сигнали для інтерпретатора і користувача. Кожен висновок нейронів будь-якого шару подається на вхід всіх нейронів наступного шару. Число нейронів в шарі може бути будь-яким.

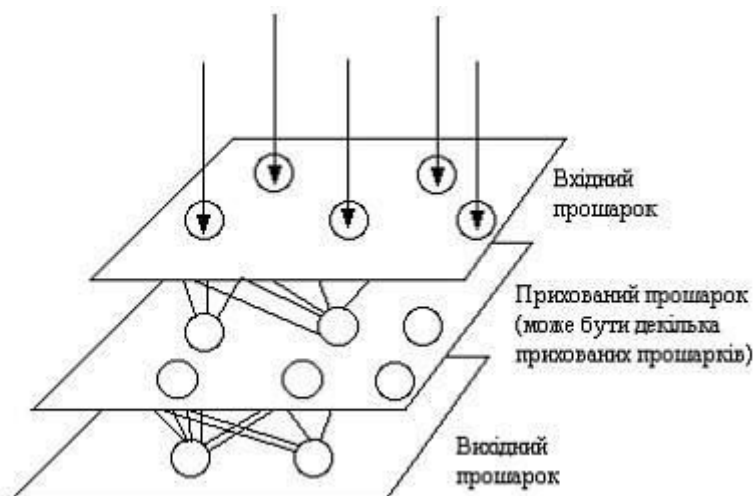


Рисунок 1.7 - штучна нейронна мережа

2. РОЗРОБКА МЕТОДУ РОЗПІЗНАВАННЯ ОБРАЗІВ ДЛЯ СИСТЕМИ КОРЕГУВАННЯ ЗОБРАЖЕНЬ

2.1 Уявлення зображень

Розпізнавання образів – наукова дисципліна, метою якої є виявлення об'єктів за кількома критеріями або класами. Теорія розпізнавання об'єктів являє собою розділ інформатики, який ґрунтується на розробці основ і методів ідентифікації предметів, явищ і сигналів. Потреба в такому розпізнаванні виникає в багатьох областях, починаючи з машинного зору, символного розпізнавання, діагностики в медицині, розпізнавання мови і закінчуючи вузько спеціальними завданнями. Незважаючи на те, що деякі з цих завдань вирішуються людиною на підсвідомому рівні з великою швидкістю, до теперішнього часу ще не створено комп'ютерних програм, вирішальних їх в настільки ж загальному вигляді. У зв'язку з цим, проблема розпізнавання образів отримала повсюдне поширення, в тому числі в галузі штучного інтелекту і робототехніки.

Можливість розпізнавання базується на схожості подібних об'єктів. Незважаючи на те, що всі явища і предмети не схожі один на одного, між деякими з них завжди можна знайти подібності з того чи іншою ознакою. Всі методи розпізнавання об'єктів діляться на два види: методи, засновані на теорії рішень і структурні методи. Перші засновані на обчисленні за допомогою кількісних величин, таких як довжина, текстура і т.д. Другі орієнтовані на образи, для опису яких більше підходять якісні величини, наприклад реляційні. Також в розпізнаванні об'єктів важливу роль відіграє навчання на основі відомої вибірки. Під чином мається на увазі деяка упорядкована сукупність ознак. Класом образів називається сукупність об'єктів з однаковими властивостями. Класифікатором або вирішальним

правилом називається правило віднесення образу до одного з класів на підставі його вектора ознак.

На практиці широке застосування мають три форми подання ознак: вектор ознак (для кількісних величин), символічний рядок і дерева ознак (для структурних величин). Методи, засновані на зіставленні, є наборами векторів ознак кожного класу об'єктів. Новий образ буде віднесений до того класу, який виявиться найбільш близьким, в межах заздалегідь заданій метрики. Очевидно, що найпростіший підхід полягає в пошуку мінімального відстані, яке обчислюється за допомогою евклідових норм між векторами ознак невідомого об'єкта і векторами прототипу. Висновок про належність об'єкта до певного класу відбувається за найменшим з цих відстаней. Мінімальний класифікатор відстані добре працює в тих випадках, де відстань між точками математичного очікування класів велике в порівнянні з діапазоном розкиду об'єктів кожного класу.

Не менш важливими є методи розпізнавання образів, засновані на ймовірнісних класифікаторах, через випадковостей, які впливають на породження класів образів. Отже, необхідно виробити такий оптимальний підхід, при використанні якого виявиться найменша ймовірність появи помилок. Дуже складно однозначно відповісти, як виглядає оптимальний метод описує комп'ютерне зір. Однак, можна розділити всі існуючі методи на три ступені: первинна обробка і фільтрація, логічна оцінка результатів фільтрації і алгоритми прийняття рішень.

Як правило, для розпізнавання об'єктів на зображенні необхідно застосувати всі ці етапи, однак буває достатньо двох, або навіть одного. До групи фільтрації можна віднести методи, які дозволяють визначити на зображенні цікавлять об'єкти, без попереднього аналізу. Основна маса таких методів використовує будь-яку єдину операцію на всі точки зображення одночасно. На даному рівні аналіз як правило не проводиться.

Найпростішим перетворенням є бінаризація зображення по порогу. Для зображень RGB і в градаціях сірого таким порогом є значення яскравості. Вибір порога, що визначає бінаризація, визначає вид самого процесу. Як правило, бінаризація відбувається при алгоритмі адитивного вибору порога. Наприклад, таким алгоритмом може стати вибір математичного очікування або моди, а також найбільшого піку гістограми. Існуючі класичні методи фільтрації можуть бути застосовані в широкому спектрі завдань. Найбільш поширеним класичним методом є перетворення Фур'є, проте він не використовується в зображеннях в чистому вигляді. Однак для аналізу зображень часто буває недостатньо простого одновимірного перетворення, і потрібно набагато більше ресурсомісткий двовимірне перетворення:

$$G_{uw} = \frac{1}{NM} \sum_{n=1}^{N-1} \sum_{m=1}^{M-1} x_{mn} e^{-2\pi j \left[\frac{wu}{M} + \frac{wu}{N} \right]} \quad (2.1)$$

Обчислення за такою формулою є досить трудомістким, тому на практиці частіше користуються згорткою цікавить області за допомогою низькочастотних або високочастотних фільтрів, в залежності від конкретного завдання. Таке спрощення звичайно, не дозволяє більш широкого діапазону операцій, таких як аналіз, однак найчастіше буває досить тільки результату без подальших перетворень. Вейвлет-перетворення є більш перспективним і сучасним методом обробки зображень, ніж перетворення Фур'є. Вони спрощують стиснення, аналіз і передачу великої кількості зображень. Вейвлет-перетворення засновані на розкладанні по малим хвилях (вейвлет) з частотою, що змінюється і

обмеженням за часом, на відміну від перетворення Фур'є, побудованого на гармонійних функціях.

У 1987 році Стефан Маллат вперше продемонстрував, що вейвлети можуть бути покладені в основу принципово нового методу обробки зображень, що отримав назву кратномасштабного аналізу. Як очевидно з назви, кратномасштабна теорія має справу з аналізом зображень при різних дозволах, так як багато деталей, непомітні при одному масштабі, можуть бути легко знайдені при іншому. Довгий час вейвлети мали досить обмеженим поширенням, проте зараз вже важко встежити за всією інформацією, наявною по цій темі. При погляді на зображення, ми бачимо пов'язані набори об'єктів однакової яскравості і структури, які об'єднуючись утворюють предмети або області відображення. Коли присутні одночасно, як маленькі об'єкти, так і великі, то аналіз зображення в різних дозволах дозволить значно розширити області обробки.

З математичної точки зору зображення є двовимірною матрицею значень яскравості. Однак при переході від однієї його частини до іншої, навіть такі статистики першого порядку, як гистограми значно змінюються. Існує набір класичних функцій, що застосовуються в вейвлет перетвореннях: вейвлет Хаара, вейвлет Морлі, вейвлет Добеши і т. Д. Хорошим прикладом застосування вейвлет аналізу є завдання пошуку відблиску в зіниці ока, де вейвлетом є сам відблиск. В основі вейвлетів лежить кореляція, яка може застосовуватися як в сукупності з іншими методами, так і самостійно. При розпізнаванні образу в зображенні це незамінний інструмент. Іншим не менш цікавим класом фільтрації є фільтрація функцій. Вона дозволяє на простому зображенні знайти безліч шматочків найпростіших функцій (пряма, парабола і т. Д.).

Найбільш відомим є перетворення Хафа, яке дозволяє знаходити будь-які ефективно обчислювальної функції. Його аналогом є

перетворення Радону, яке за рахунок обчислення через швидке перетворення Фур'є дає вигоду в продуктивності. Окремий розділ фільтрації - фільтрація контурів. Вона дуже корисна в тій ситуації, коли об'єкт досить складний, але має чіткі межі. Тоді фільтрація контурів є мало не одним з основних інструментів роботи з зображенням та проводиться з використанням операторів Кенні, Лапласа, Прюїтт, Собеля і Робертса. Розглянуті фільтри можуть вирішити більшість завдань, проте не варто забувати про менш поширених, але які використовуються в локальних завданнях: ітераційні фільтри, курвлет і бамблет перетворення і т. Д. Поле фільтрації на виході виходить набір даних піддаються обробці. Але часом вони все ж вимагають додаткових логічних перетворень. Тому необхідне введення методів, що дозволяють перейти від цілого зображення до властивостей об'єктів на ньому.

Методи математичної морфології є засобом переходу від фільтрації до логіки. Вони дозволяють прибрати шуми на бінарному зображенні, змінивши розмір наявних елементів. Також існує безліч методів, які дозволяють ідентифікувати об'єкт по контуру. Такий підхід називається контурним аналізом. Особливі точки є унікальними характеристиками які дозволяють зіставити різні класи об'єктів. Існує три види особливих точок: особливі точки, які є стабільними протягом часу; особливі точки, які є при зміні освітлення і невеликих рухах об'єкта; і стабільні особливі точки.

Методи машинного навчання і прийняття рішень є фінальною стадією в розпізнаванні образів. Вони знаходяться на стику математичної статистики, методів оптимізації та класичних математичних дисциплін, але має також і власну специфіку, пов'язану з проблемами обчислювальної ефективності та перенавчання. У більшості випадків суть навчання полягає в наступному: на основі навчальної вибірки з ознаками кожного класу побудувати таку модель, за допомогою якої машина зможе проаналізувати

нове зображення і вирішити, який з об'єктів є на зображенні. Існує два типи навчання: на основі людських знань, перенесених в комп'ютер у вигляді бази і навчання по прецедентах (індуктивне) засноване на виявленні закономірностей.

У реальних прикладних задачах вхідні дані про об'єкти можуть бути неповними, неточними, нечисловими, різноманітними. Ці особливості призводять до великої різноманітності методів машинного навчання. Таким чином, ми провели короткий аналіз існуючих методів машинного розпізнавання образів. Штучний інтелект і деякі суміжні області, такі як аналіз сцен і машинне зір, все ще перебувають на початкових стадіях розвитку. Однак, описані підходи дійсно вкрай різноманітні і за допомогою більшості з них можна вирішити практично будь-яке завдання розпізнавання образів.

2.2 Редагування на рівні людини: Крива

Інструмент кривих є, мабуть, найпотужнішим і гнучким засобом перетворення зображень. Інструмент кривих може приймати вхідні тони і вибірково розтягувати або стискати їх. Однак, на відміну від рівнів, які мають тільки чорний, білий і середній контроль, тональна крива контролюється з використанням будь-якої кількості опорних точок. Результат даної кривої можна візуалізувати, слідуючи тестовому вхідного тонального сигналу до кривої, потім переходячи до її результуючому вихідному тонального сигналу. Тому діагональна лінія, що проходить через центр, не змінить тонів. У наших експериментах ми модифікуємо координати $(x \text{ і } y)$ N точок на кривій, які відповідають діям політики a .

2.3 Policy Gradient

Для кожної фотографії ми виберемо кілька безперервних дій $a = \{a_1, a_2 \dots a_n\}$, які відповідають точкам на кривій, згаданим в попередньому розділі. Модель являє собою стохастическую політику $p_i(a | x)$, де x – вхідна фотографія. Тут ми моделюємо криву як безперервну завдання управління. Кожна дія являє собою значення μ гауссовского розподілу. Щоб зменшити простір дії, ми фіксуємо сигму на константу. Під час навчання ми будемо пробувати дії з дистрибутива. Після отримання дій ми опрацюємо фотографію на основі a , яка утворює породжує модель $G(y | x)$, де y – оброблена фотографія з кривою, згаданої в попередньому розділі. Ми безпосередньо максимізували винагороду $R(a)$.

2.4 Дискриминатор для змагального навчання

Наша модель отримує винагороду від параметризовані дискримінаційної моделі D (Goodfellow і ін. 2014 року). D – це ймовірність, що показує, наскільки ймовірно, що фотографія добре оптимізована, іншими словами, шедевр, подібний фотографії. Дискримінуюча модель D навчається шляхом надання позитивних прикладів S з фотографії з високим рейтингом і фотографії, створеної мережею політики.

2.5 Навчання змагальності з Policy Gradient

Модель генератора (політика) $G(y | x)$ полягає в тому, щоб згенерувати фотографію y з необробленого x , щоб максимізувати

очікуване винагороду, де R – нагорода за фотографію. Зверніть увагу, що винагорода виходить від дискримінатора D . $Q(x, y)$ є функцією значення дії фотографії, тобто очікуваного винагороду, починаючи з x , виконуючи дію a , а потім слідуючи політиці G . Тут ми розглядаємо оціночну ймовірність бути шедевром дискримінатором $D(y)$ в якості нагороди.

Перевага використання дискримінатора D в якості опції винагороди полягає в тому, що він може динамічно оновлюватися для подальшого ітеративного поліпшення генеративної моделі. Це також служить нагородою сприйняття, що означає, що ми можемо отримати дані тренувань з фотографій в дикій природі.



Рисунок 2.1 – Тренування людини

Методи градієнта політики (PG) є часто використовуваними алгоритмами в навчанні з підкріпленням (RL). Принцип дуже простий. Ми спостерігаємо і діємо. Людина вживає заходів, засновані на

спостереженнях. Як цитата з Стівена Каррі: ви повинні покладатися на той факт, що ви вкладаєте роботу в створення м'язової пам'яті, а потім вірите, що вона працює. Причина, по якій ви так багато тренується і працюєте над цим, полягає в тому, що під час гри ваші інстинкти переходять в точку. де це дивно, якщо ви не робите це правильно.

Постійне тренування – це ключ до нарощування м'язової пам'яті у спортсменів. Для PG ми розробляємо політику, засновану на спостереженнях. Навчання в PG робить дії з високими нагородами більш ймовірними, або навпаки.

Ми зберігаємо те, що працює, і викидаємо те, що немає.

В градієнтах політики, Каррі є нашим агентом.

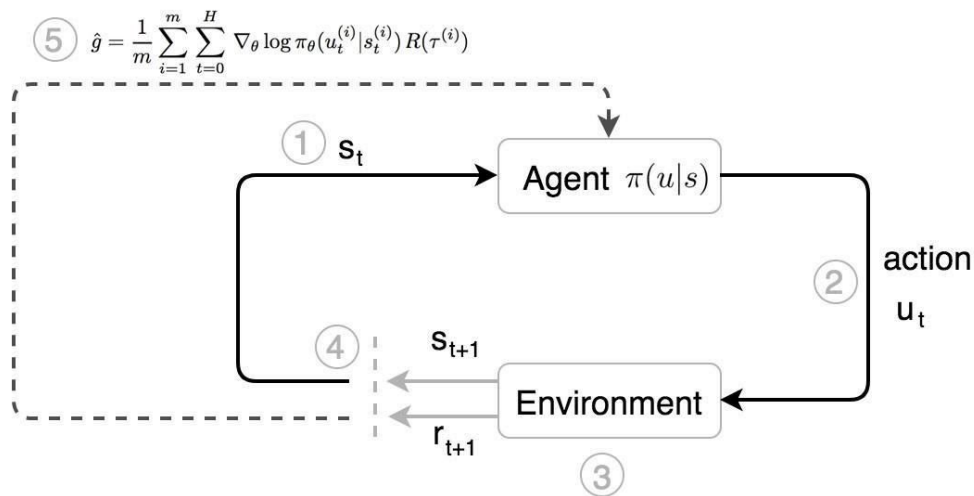


Рисунок 2.2 – Градієнт політики

Він спостерігає за станом навколишнього середовища.

Він робить дію (u), ґрунтуючись на своєму інстинкті (політиці π) щодо стану s .

Він рухається, і противники реагують. Нова держава формується.

Він робить подальші дії в залежності від спостережуваного стану.

Після траєкторії τ рухів він налаштовує свій інстинкт на основі сумарного винагороди $R(\tau)$, отриманого.

Каррі візуалізує ситуацію і миттєво знає, що робити. Роки навчання вдосконалюють інстинкт, щоб максимізувати винагороду. В RL інстинкт може бути математично описаний як:

$$\pi(u|s) \quad (2.2)$$

Ймовірність вчинення дії з урахуванням стану s . π є політикою в RL. Наприклад, наскільки ймовірним є повороту або зупинки, коли ви бачите автомобіль попереду:

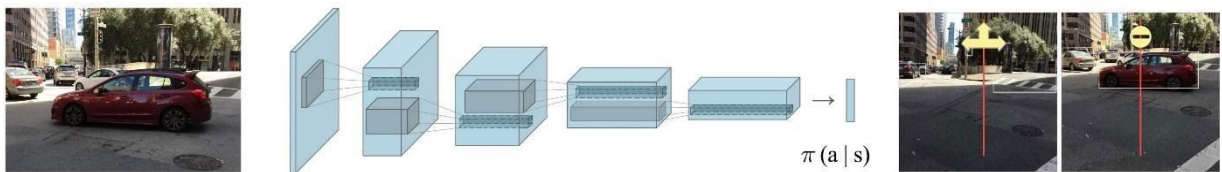


Рисунок 2.3 – Приклад

Як ми можемо сформулювати нашу мету математично? Очікувані винагороди дорівнюють сумі ймовірності траєкторії \times відповідних винагород:

$$J(\theta) = \mathbb{E} \left[\sum_{t=0}^H R(s_t, u_t); \pi_{\theta} \right] = \sum_{\tau} P(\tau; \theta) R(\tau) \quad (2.3)$$

І наша мета полягає в тому, щоб знайти політику θ , яка створює траєкторію τ

$$(s_1, u_1, s_2, u_2, \dots, s_H, u_H) \quad (2.4)$$

це максимізує очікувані нагороди.

$$\max_{\theta} J(\theta) = \max_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \quad (2.5)$$

Вхідні функції і нагороди

$$\pi_{\theta}(u|s) \quad (2.6)$$

Це можуть бути елементи ручної роботи для стану (наприклад, кути з'єднання / швидкість маніпулятора), але в деяких проблемних областях RL досить зрілий, щоб безпосередньо обробляти необроблені зображення. π може бути детермінованою політикою, яка виводить точне заходів, яких треба вжити (перемістіть джойстик вліво або вправо). π також може бути стохастичною політикою, яка виводить можливість дії, яке воно може зробити.

Ми записуємо винагороду, дане на кожному часовому кроці. У баскетбольному матчі усі рівні 0, крім стану завершення, рівного 0, 1, 2 або 3.

$$(s_1, u_1, r_1, s_2, u_2, r_2, \dots, s_H, u_H, r_H) \quad (2.7)$$

Давайте введемо ще один термін H , званий горизонтом. Ми можемо нескінченно запускати курс моделювання ($h \rightarrow \infty$), поки він не досягне кінцевого стану, або ми не встановимо обмеження на H кроків.

2.6 Оптимізація

По-перше, давайте визначимо загальний і важливий трюк в Deep Learning і RL. Приватна похідна функції $f(x)$ (RHS) дорівнює $f(x)$, помноженої на приватну похідну від $\log(f(x))$.

$$f(x) \nabla_{\theta} \log f(x) = f(x) \frac{\nabla_{\theta} f(x)}{f(x)} = \nabla_{\theta} f(x) \quad (2.8)$$

Замінити $f(x)$ на π .

$$\pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau) = \nabla_{\theta} \pi_{\theta}(\tau) \quad (2.9)$$

Крім того, для безперервного простору очікування може бути виражено як:

$$E_{x \sim p(x)} [f(x)] = \int p(x) f(x) dx \quad (2.10)$$

Тепер давайте математично формалізуємо нашу проблему оптимізації. Ми хочемо змоделювати політику, яка створює траєкторії, які максимізують загальну винагороду.

$$\theta^* = \arg \max_{\theta} \underbrace{E_{\tau \sim p_{\theta}(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]}_{J(\theta)} \quad (2.11)$$

Однак, щоб використовувати градієнтний спуск для оптимізації нашої задачі, чи потрібно нам брати похідну функції винагороди r , яка не може бути дифференцируемой або формалізованої?

Давайте перепишемо нашу цільову функцію J як:

$$J(\theta) = E_{\tau \sim \pi_{\theta}(\tau)}[r(\tau)] = \int \pi_{\theta}(\tau) r(\tau) d\tau \quad (2.12)$$

Градієнт (градієнт політики) стає:

$$\begin{aligned} \nabla_{\theta} J(\theta) &= \int \nabla_{\theta} \pi_{\theta}(\tau) r(\tau) d\tau = \int \pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau) r(\tau) d\tau \\ &= E_{\tau \sim \pi_{\theta}(\tau)}[\nabla_{\theta} \log \pi_{\theta}(\tau) r(\tau)] \end{aligned} \quad (2.13)$$

Градієнт політики може бути представлений як очікування. Це означає, що ми можемо використовувати вибірку, щоб наблизити її. Крім того, ми вибираємо значення r , але не диференціюючи його. Це має сенс, тому що винагорода не залежить безпосередньо від того, як ми параметризуємо модель. Але траєкторії τ є. Отже, яка приватна похідна від $\log \pi(\tau)$.

$\pi(\tau)$ визначається як:

$$\underbrace{\pi_\theta(\mathbf{s}_1, \mathbf{a}_1, \dots, \mathbf{s}_T, \mathbf{a}_T)}_{\pi_\theta(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^T \pi_\theta(\mathbf{a}_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) \quad (2.14)$$

Візьми журнал:

$$\log \pi_\theta(\tau) = \log p(\mathbf{s}_1) + \sum_{t=1}^T \log \pi_\theta(\mathbf{a}_t | \mathbf{s}_t) + \log p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) \quad (2.15)$$

Перше і останнє доданок не залежить від θ і може бути видалено.

$$\nabla_\theta \left[\cancel{\log p(\mathbf{s}_1)} + \sum_{t=1}^T \log \pi_\theta(\mathbf{a}_t | \mathbf{s}_t) + \cancel{\log p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)} \right] \quad (2.16)$$

Таким чином, політика градієнта

$$\nabla_\theta J(\theta) = E_{\tau \sim \pi_\theta(\tau)} [\nabla_\theta \log \pi_\theta(\tau) r(\tau)] \quad (2.17)$$

буде виглядати так:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \right) \left(\sum_{t=1}^T r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right) \quad (2.18)$$

І ми використовуємо цей градієнт політики для оновлення політики.

2.7 Інтуїція

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \right) \left(\sum_{t=1}^T r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right) \quad (2.19)$$

Як ми можемо зрозуміти ці рівняння? Підкреслений термін – максимальна логарифмічна ймовірність. При глибокому навчанні він вимірює вірогідність спостережуваних даних. У нашому контексті він вимірює, наскільки ймовірна траєкторія відповідно до поточної політики. Помноживши це на винагороду, ми хочемо збільшити ймовірність політики, якщо траєкторія призведе до високого позитивного винагороді. Навпаки, ми хочемо зменшити ймовірність політики, якщо вона призведе до високої негативної нагороди. Коротше, продовжуй, що працює, і викидай, що немає.

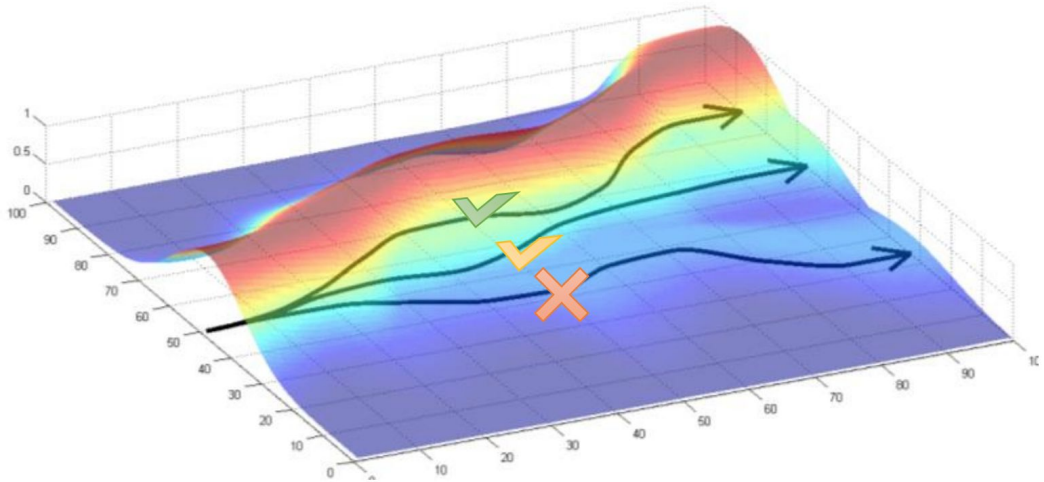


Рисунок 2.4 – Візуалізація прикладу

Якщо піднятися в гору нижче означає більш високі винагороди, ми змінимо параметри моделі (політику), щоб збільшити ймовірність траєкторій, які рухаються вище.

У градієнті політики є одна важлива річ. Ймовірність траєкторії визначається як:

$$\pi_{\theta}(\tau) = p(\mathbf{s}_1) \prod_{t=1}^T \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) \quad (2.20)$$

Стану в траєкторії сильно пов'язані між собою. У глибокому навчанні довга послідовність множення з сильно корельованими факторами може легко викликати зникнення або вибух градієнта. Однак градієнт політики тільки підсумовує градієнт, який руйнує прокляття множення довгою послідовності чисел.

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \right) \left(\sum_{t=1}^T r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right) \quad (2.21)$$

Як результат:

$$\pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau) = \nabla_{\theta} \pi_{\theta}(\tau) \quad (2.22)$$

3. РОЗРОБКА СИСТЕМИ КОРЕГУВАННЯ ЗОБРАЖЕНЬ

3.1 Особливості та переваги використання мови програмування Python

Python проста у використанні, та водночас повноцінна мова програмування, що надає набагато більше засобів для структурування і підтримки великих програм, ніж shell. З іншого боку, вона краще за C обробляє помилки, і, будучи мовою дуже високого рівня, має вбудовані типи даних високого рівня, такі як гнучкі масиви і словники, ефективна реалізація яких на C потребує значних витрат часу.

Завдяки більш загальним типам даних, Python застосовують до більш широкого кола задач, ніж Awk і навіть Perl, у той ж час багато речей на мові Python робляться настільки ж просто.

Python дозволяє розбивати програми на модулі, що потім можуть бути використані в інших програмах. Python поставляється з великою бібліотекою стандартних модулів, які можна використовувати як основу для нових програм або як наприклад при вивченні мови. Стандартні модулі надають засоби для роботи з файлами, системними викликами, мережевими з'єднаннями і навіть інтерфейсами до різних графічних бібліотек.

Python – інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови, писати шаблони програм або тестувати функції при розробці «знизу-вверх». Він також зручний як настільний калькулятор. Python дозволяє писати дуже компактні й зручні для читання програми.

Програми, написані мовою Python, звичайно значно коротші еквівалента на C або C++ з декількох причин:

- типи даних високого рівня дозволять Вам виразити складні операції однією інструкцією;
- групування інструкцій виконується за допомогою відступів замість фігурних дужок;
- немає необхідності в оголошенні змінних.

Python розширювана мова: знання C дозволяє додавати нові функції, що вбудовуються, або модулі для виконання критичних операцій з максимальною швидкістю або написання інтерфейсу до комерційних бібліотек, доступним тільки у двійковій формі. Інтерпретатор мови Python може бути вбудований у програму, написану на C, і використовувати його як розширення або командну мову для цієї програми. Python використовується в даний час десятками тисяч програмістів в усьому світі, і число людей, що використовують його, швидко зростає, подвоюється і потроюється щороку. Python приваблює користувачів з ряду причин. Він використовується для розробки програм і дозволяє провести розробку набагато швидше, ніж традиційні мови типу C, C++ або Java . Ця мова працює однаково добре на Windows, UNIX, Macintosh, і OS/2, може використовуватися, для легкої розробки як малих додатків чи сценаріїв, так і для розгортання великих програм. Python пропонує доступ до могутнього і легкого у використанні комплекту 29 інструментальних засобів графічного інтерфейсу користувача. Традиційні машинні мови типу C і Pascal мають ряд характеристик, наприклад, суворі типізація, базові типи, складні (і звичайно довгі) цикли, і потреба у великих кількостях кодів для виконання відносно малих задач. Java досить новий, але розділяє більшість характеристик, включених у цей перелік.

Програмісти, знайомі з традиційними мовами погодяться, що відсутність суворої типізації полегшує роботу з Python.

3.2 Python у порівнянні з іншими мовами

Відмінностей Python від інших мов доволі багато, перерахуємо основні з них:

- керування пам'яттю – цілком автоматичне – не потрібно хвилюватися щодо розподілу або звільнення пам'яті. Немає загрози «небезпечного посилання». Java - єдина мова, що пропонує таку концепцію;

- типи зв'язані з об'єктами, а не зі змінними. Це означає, що змінній може бути призначене значення будь-якого типу, і що (наприклад) масив може містити об'єкти різних типів. Традиційні мови не надають такої можливості;

- операції звичайно виконуються в більш високому рівні абстракції. Це частково результат того, як написана мова, і частково результат розширеної стандартної бібліотеки кодів, що поставляється разом з Python.

Ці та інші особливості Python роблять розгортання додатків надзвичайно швидким. Продуктивність створеного додатку залежить від його особливостей. Звичайно, для чисельного алгоритму, що виконує звичайну арифметику цілого числа в циклі 'for', неважливо, на якій мові він написаний. Але для «середнього» додатка, збільшення продуктивності може бути просто дивовижним.

Один недолік Python, у порівнянні з найбільш традиційними мовами, полягає в тому, що це – не цілком компільована мова; замість цього, вона частково трансліює програму до внутрішньої форми байт-коду, і цей

байт-код виконується інтерпретатором Python. Однак, у перспективі – сучасні комп'ютери мають так багато не використуваного обчислювального потенціалу, що для 90% додатків швидкодія зв'язана з вибором мови. Java теж компілюється в байт-код, але в даний час працює повільніше ніж Python у більшості випадків. Крім того, дуже просто об'єднати Python з модулями, написаними на C або C++, які можна використовувати, щоб збільшити швидкість роботи програм в критичних ділянках.

3.3 Розробка програми

У результаті огляду існуючих мов програмування були наведені особливості та переваги вибору мови програмування, у результаті було обрано Python. PyCharm був обраний в якості середовища розробки. Також використовувалось бібліотеки Policy, CNN, Tensorflow. Було розроблене програмне забезпечення у вигляді API.

API здатне приймати тестову вибірку а також отпрацьовувата окремі зображення після проведення навчання на такій виборці.

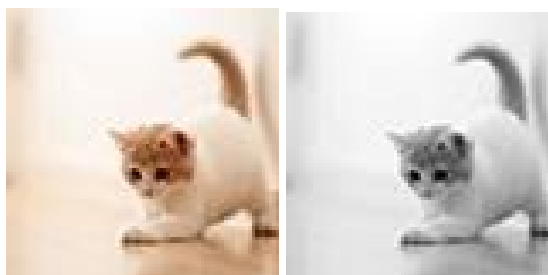


Рисунок 3.1 – Результат виконання програми

Після проходження тренування на тестовій виборці програма здатна приміняти такі саме корекції на нове зображення, як ті, що були використані для корекції зображень у тестовій виборці.

На рисунку 3.1 наведено приклад виконання програми після тренування на виборці з зображень, що були скориговані до чорно-білого стану. Текст програми наведено у додатку А.

Данне АРІ можливо інтегрувати у програми, які працюють з великою кількістю зображень. Так, наприклад, редактор зображень зможе за допомогою даного АРІ після тренувань на виборці з корекцій зображень користувача зробити автоматичну корекцію нових зображень відповідно до попередніх коригувань цього користувача з урахуванням його уподобань.

ВИСНОВКИ

У даному дипломному проекті розроблена система, працююча на основі нейронної мережі в системі корегування зображень. У результаті проведеного аналізу предметної області не було знайдено схожих систем на відкритому ринку та проаналізовані існуючі методи, які не зовсім задовольняли таку задачу. У результаті огляду існуючих мов програмування були наведені особливості та переваги вибору мови програмування, у результаті було обрано Python. PyCharm був обраний в якості середовища розробки. Також використовувалось бібліотеки Policy, CNN, Tensorflow. Було розроблене програмне забезпечення.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

- 1 Що таке Python? [Електронний ресурс]: Режим доступа: URL: <http://www.plug.org.ua/documentation/about-python> (дата звернення- 10.03.2020)
- 2 Policy Gradient. [Електронний ресурс]: Режим доступа: URL: http://www.scholarpedia.org/article/Policy_gradient_methods (дата звернення- 13.02.2020)
- 3 Метод ветвей и границ. [Електронний ресурс]: Режим доступа: URL: <http://ru.wikipedia.org/wiki> (дата звернення- 13.02.2020)
- 4 Рейнгольд Э. Комбинаторные алгоритмы. Теория и практика. - М.: Мир, 1980. — 476 с.
- 5 Пушкарёва Г.В. Исследование и применение бионических методов и моделей для автоматизированного проектирования маршрутов обхода геометрических объектов Компьютерная графика и представление GraphiCon 2005: науч.-техн. конф., 20-24 июня 2005г. : - 2005. - Режим доступа: URL: <http://www.graphicon.ru/2005/proceedings/papers/Pushkaryova.pdf> (дата звернення- 14.02.2020)
- 6 Генетический алгоритм. [Електронний ресурс]: Режим доступа: URL: <http://ru.wikipedia.org/wiki> (дата звернення- 20.02.2020)
- 7 Обобщённая задача коммивояжёра для определения рациональных маршрутов поставки. [Електронний ресурс]: Режим доступа: URL: <http://econference.ru> (дата звернення- 22.02.2020)
- 8 Glover F. Tabu Search Journal of the Operational Research Society. - 1999. – Vol.50, № 1. – pp. 106–107. - Режим доступа: URL: <http://glossary.computing.society.informs.org/notes/spanningtree.pdf> (дата звернення- 25.02.2020)

9 Сухарев А.Г. Курс методов оптимизации: Учебное пособие. - [2-е изд]. – М.: ФИЗМАТЛИТ, 2005. – 368 с.

10 Муравьиный алгоритм. [Электронный ресурс]: Режим доступа: URL: <http://ru.wikipedia.org/wiki> (дата звернення- 01.03.2020)

11 Штовба С.Д. Муравьиные алгоритмы// Exponenta Pro. Математика в приложениях. - 2003. – № 4. – С. 70-75

12 СТП 1-У-НГТУ-98 «Проекты (работы) дипломные и курсовые. Общие требования к оформлению пояснительных записок и чертежей"