НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
В.о. завідувача кафедри

_____ М.В.Грайворонський
(підпис)

“_____”_____

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему: Захист від атак на нейронні мережі, що вирішують задачу класифікації зображень_____

Виконав: студент _4__ курсу, групи __ФБ-52_____

Могир Максим Сергійович_____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник_____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

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

Студент _____
(підпис)

Київ - 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____
М.В.Грайворонський
«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту
Могиру Максиму Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: «Захист від атак на нейронні мережі, що вирішують задачу класифікації зображень»,

науковий керівник роботи: доц. каф. ІБ, к.т.н., Родіонов Андрій Миколайович,

затверджені наказом по університету від «27» травня 2019 р. № 1414-с

2. Термін подання студентом роботи: 10 червня 2019

3. Вихідні дані до роботи: методи машинного навчання для класифікації зображень та алгоритми захисту нейронних мереж

4. Зміст роботи:

- Огляд існуючих атак на нейронні мережі
- Аналіз існуючих алгоритмів захисту нейронних мереж
- Побудова власної змагальної нейронної мережі, що буде використовуватись для захисту класифікатору

- Аналіз результатів і порівняння з існуючими методами

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

- Презентація

6. Дата видачі завдання: 18 вересня 2018 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Вивчення літератури за тематикою проекту	18.09.18 – 21.11.18	
2	Аналіз існуючих атак на нейронні мережі	21.11.18 – 10.12.18	
3	Написання першого розділу	10.12.18 – 23.12.18	
4	Аналіз існуючих алгоритмів захисту нейронних мереж	23.01.19 – 20.02.19	
5	Написання другого розділу	20.02.19 – 1.03.19	
6	Розробка власного методу і написання програмної частини	1.03.19 – 1.04.19	
7	Написання третього розділу	1.04.19 – 25.04.19	
8	Проходження переддипломної практики	15.04.19 – 14.05.19	
9	Оформлення висновків, джерел	14.05.19 – 20.05.19	
10	Передзахист проекту	20.05.19 – 30.05.19	
11	Підготовка графічної частини	02.06.19-04.06.19	
12	Захист дипломної роботи	20.06.19	

Студент _____ Могир М. С.

Науковий керівник роботи _____ Родіонов А.М.

РЕФЕРАТ

Робота обсягом 60 сторінок містить 25 ілюстрації, 3 таблиці, 17 літературних посилань.

В роботі мною було розглянуто атаки на штучні нейронні мережі, що виконують завдання класифікації зображень, методи захисту від таких атак, та була проведена порівняльна характеристика останніх напрацювань в даній сфері. Також були досліджені атаки на різні архітектури нейронних мереж, їх результати класифікації зображень,як зовсім без захисту,так і з застосованими алгоритмами для зменшення впливу атаки на результат мережі. На основі даних досліджень було написано алгоритм,що будується на змагальних штучних нейронних мережах, який допомагає моделям боротись з атаками, направленими на погіршення результатів класифікації.

Результати можна використати у всіх системах, що займаються класифікацією зображень,і використовують для цього нейронні мережі. Це покращить результати класифікації, та допоможе автоматично позбутись впливу атак на точність класифікатору.

НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, АТАКИ НА НЕЙРОННІ МЕРЕЖІ, АЛГОРИТМИ ЗАХИСТУ,ЗМАГАЛЬНІ НЕЙРОННІ МЕРЕЖІ.

ABSTRACT

The work of 60 pages contains 25 illustrations,3 tables,17 literary references.

In my work, I considered attacks on artificial neural networks that perform tasks of image classification, methods for protecting against such attacks, and a comparative characteristic of recent developments in this area was carried out. Attacks on various neural network architectures, their results of image classification as completely unprotected, and with the applied algorithms to reduce the impact of attack on the result of the network were also investigated. Based on research data, an algorithm based on competitive artificial neural networks was written that helps the models fight off attacks aimed at worsening the classification results.

The results can be used on all systems involved in the classification of images, and use neural networks for this purpose. This will improve the results of the classification, and will help automatically get rid of the impact of attacks on the accuracy of the classifier.

NEURAL NETWORKS, IMAGE CLASSIFICATION, ATTACKS ON NEURAL NETWORKS, PROTECTION ALGORITHMS, ADVERSARIAL NEURAL NETWORKS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

MSE – Mean Squared Error (середня квадратична помилка)

lr – learning rate (швидкість навчання)

GAN – Generative Adversarial Network (генеративна змагальна мережа)

ResNet – Residual Network (залишкова мережа)

VGGNet – Visual Geometry Group Network

ЗНМ – змагальна нейронна мережа

ВСТУП

Сьогодні, у час розвитку алгоритмів машинного навчання, є надзвичайно важливим забезпечення нормальних умов для їх роботи. За останні роки, глубокі нейронні мережі були широко використовувані для задач машинного навчання, включаючи класифікацію. Однак вони виявились вразливими до певного роду атак: невеликі зміни картинки можуть спричинити помилкову класифікацію потрібних нам зображень[1]. Наскільки стрімко розвиваються алгоритми комп'ютерного зору, штучні нейронні мережі, згорткові нейронні мережі, настільки ж швидко зловмисники намагаються придумати способи завадити їх роботі, та змінити їх результати для своєї вигоди.

Декілька компаній в світі займається розробкою та виготовленням безпілотних автомобілів, які здатні отримувати інформацію про середовище, та рухатись дорогами без втручання людини. Багато з них використовують нейронні мережі у більшості своїх завдань – починаючи від отримання загальної картини середовища, закінчуючи розпізнаванням дорожніх знаків та сигналів від інших водіїв[2]. Якщо уявити собі, що якась невеличка наклейка на дорожньому знаку може повністю зламати результат моделі, що відповідає за розпізнавання, стає зрозуміло настільки важлива ця проблема, навіть в рамках однієї області використання нейронних мереж.

Варто згадати і про системи, що використовують модулі розпізнавання облич. Вони виконують завдання автоматизації ідентифікації особистості по відео або фото[3]. Такі системи часто працюють як єдиний етап ідентифікації, тому вони дуже вразливі до взлому, завдяки атакам на штучні нейронні мережі. А поскільки такі моделі використовуються у багатьох системах, то доводиться вводити нові методи ідентифікації, або ж зовсім відмовлятись від розпізнавання.

Тому з метою запобігання таким втручанням у роботу нейронних мереж, пропонується використовувати певні механізми захисту, що допоможуть мережам адаптуватись до подібних атак, то протистояти всім

відомим атакам. Задля цього ми будемо використовувати нову архітектуру, яка тренується справлятись вже з зміненими зображеннями, та показувати хороші результати.

Метою даної роботи є розробка методу захисту нейронних мереж, які займаються завданням класифікації зображень, і протидія атакам, що на них спрямовані.

Для досягнення мети поставили такі завдання:

1. Аналіз існуючих атак на штучні нейронні мережі, що займаються задачами класифікації зображень.
2. Генерація прикладів атак, та розгляд причин, що можуть на це впливати.
3. Огляд структур популярних моделей нейронних мереж, що широко використовуються у різних завданнях.
4. Аналіз існуючих методів захисту від атак на такі моделі.
5. Створення моделі, що допоможе нейронним мережам адаптуватись до існуючих атак, та підвищити їх результати, під впливом атак.

Об'єкт дослідження: атаки на нейронні мережі, що вирішують завдання класифікації зображень.

Предмет дослідження: захист від атак на нейронні мережі, що вирішують завдання класифікації зображень.

Методи дослідження: для вирішення проблеми використовуються статистичні методи для аналізу результатів, та емпіричне порівняння роботи нейронних мереж.

Наукова новизна одержаних результатів в тому, що було розроблено модель яка містить в основі змагальні нейронні мережі,і може пристосовуватись до всіх типів атак, які направленні на нейронні мережі класифікації зображень.

Практичне значення одержання результатів полягає в тому, що створена модель адаптує нейронні мережі до атак, та може використовуватися у будь-яких системах, що займаються класифікацією зображень.

1 ПРОБЛЕМИ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ У АЛГОРИТМАХ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

1.1 Розвиток систем класифікації зображень та їх короткий огляд

Класифікація об’єктів є одним з найважливіших завдань в області комп’ютерного зору. Під класифікацією зображення розуміється віднесення об’єкту до однієї з можливих категорій.

У наш час класифікація зображень стала дуже важливим завданням, поскільки автоматизація такої роботи значно полегшує людям життя. Можна навести безліч прикладів, де зараз ми не можемо обійтись без класифікації: сортування фруктів та розпізнавання облич, класифікація дорожніх знаків та цифр номерних знаків автомобілів, віднесення до тих чи інших категорій природніх явищ, розпізнавання голосів, клавіатурного вводу і т.д. Щодня алгоритми комп’ютерного зору роблять мільйони виборів, відносять об’єкти до однієї чи іншої категорії, а науковці розробляють все кращі моделі, що в майбутньому замінять теперішні[4]. За останні роки алгоритми постійно змінювали один одного, але основний порядок обробки зображення залишився сталим.

Класифікація включає в себе:

- Попереднє опрацювання зображення, для того, щоб у потрібній формі подати його на вхід класифікатору.
- Знаходження на зображенні потрібного об’єкту для класифікації.
- Сегментація потрібного нам об’єкту.
- Витягнення з об’єкту певних ознак.
- Кінцевий етап,а саме – віднесення об’єкту до певної категорії.

Існує багато підходів до класифікації зображень, починаючи зі зміни попередньої обробки,закінчуючи збільшенням параметрів моделей у декілька сотень разів [5].

Один з основних підходів, що найбільш широко використовується в сфері розпізнавання об’єктів – застосування моделей-класифікаторів, що навчаються з вчителем. Для навчання таких моделей використовуються така вибірка, яка містить в собі масив зображень, і масив, що містить в собі категорії, до яких ці зображення мають бути віднесені. В процесі навчання основний масив розбивається на дві частини – вибірку для навчання, та вибірку для тестування. Після цього, використовуючи певний алгоритм, та правила його навчання, налаштовуються його параметри з використанням навчальної вибірки. Навчити ми його маємо таким чином, щоб отримавши на вхід зображення, модель на виході віддавала нам категорію, до якої відноситься клас. Даний підхід представлений багатьма моделями, серед яких найбільш широко використовувались метод опорних векторів, дерева ухвалення рішень, штучні нейронні мережі, а також ансамблі,що представляють собою суміш якихось вищеперечислених алгоритмів.

1.2 Використання для класифікації нейронних мереж

Протягом останніх років глибинне навчання показало нам, що всі стандартні алгоритми комп’ютерного зору навіть при ідеально підібраних ознаках не підберуться до результатів нейронних мереж. Вони досягали високих результатів у багатьох завданнях, не тільки у класифікації зображень. Можемо з впевненістю заявити це, дивлячись на результати маси конкурсів. Найбільш популярним і серйозним з конкурсів, в яких змагаються алгоритми комп’ютерного зору є ImageNet. Суть конкурсу полягає в тому, щоб створити таку модель, яка максимально точно класифікує заданий набір зображень[6].

Останнім часом алгоритми глубинного навчання змогли не тільки покращити результати вже існуючих моделей,не тільки підібратись дуже близько до результату у 100 відсотків, а й перевершити результат людей. Якщо трохи детальніше розповісти про ImageNet, то дані для нього були зібрані з популярних пошукових систем, вручну розмічені людьми, на 1000 категорій.

Набір даних дуже не збалансований, має багато схожих класів, які часто перетинаються або ж максимально схожі один на одного. Тому завдання складне, і вимагало нових алгоритмів, які справляться з таким викликом. На допомогу прийшли згорткові нейронні мережі, що одразу показали хороші результати, навіть на такому наборі даних. Тому майже щороку, починаючи з 2012, науковці здійснювали великі прориви у розробці глибинних моделей для задачі класифікації об’єктів на зображеннях. Завдяки такому величезному та складному набору даних ImageNet був таким собі еталоном для вимірювання роботи моделі.

Починаючи з 2014 року було створено багато масивних моделей, частини яких, трохи модернізовані, і зараз широко використовуються у різних завданнях.

1.3 Сучасні архітектури нейронних мереж для задачі класифікації.

В наш час існує декілька моделей, які змагаються за найкращий результат, та воюють вже десь на рівні сотих, або навіть тисячних відсотка. Декілька з них ми розглянемо в цьому розділі, розглянуті моделі будуть представлені в порядку представлення їх науковцями на конкурсі ImageNet.

1.3.1 VGGNet

Дана модель була створена в 2014 році, і продовжила в собі починання попередніх моделей. В ній стало ще більше шарів з персептронами, нових функцій активації, та кількості тренувальних параметрів[7].

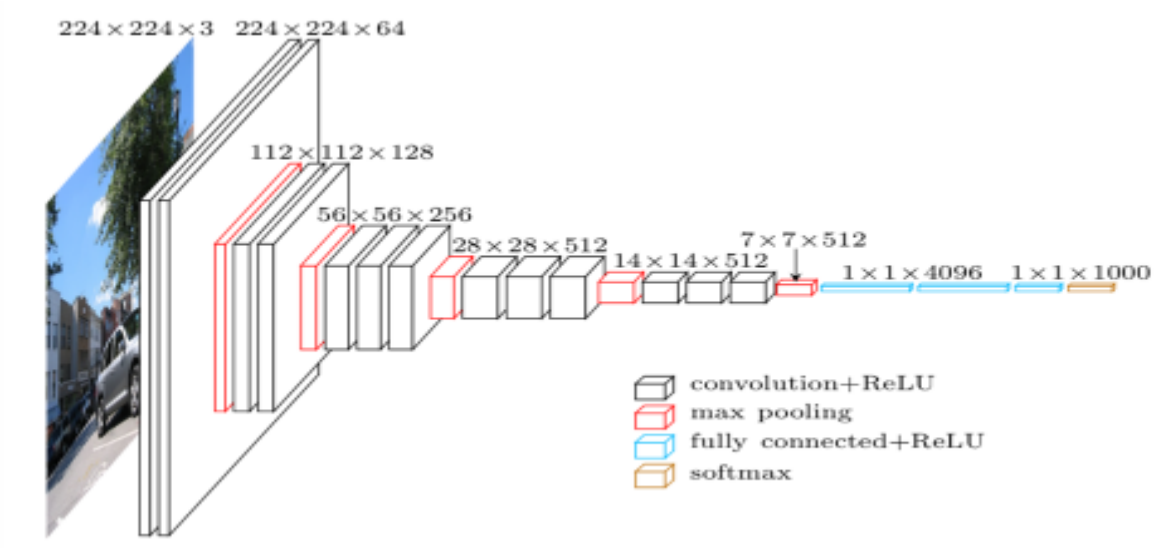

Рисунок 1.1 – Структура побудови VGGNet

На рисунку 1.1 можемо побачити через яку кількість шарів потрібно пройти зображенню, щоб отримати певну мітку класифікації. Постійно зменшуючи свої просторові розміри, зображення поступово втрачає всі непотрібні для класифікації елементи, і містить лише ті частини, які критично важливі для того, щоб модель видала правильний результат. Головна ідея в тому, що вони вирішили не використовувати якісь мудрі рішення, а просто зробили модель максимально важкою і довгою, і додали нелінійні функції активації. Порівняно з попередніми моделями були внесені такі зміни: використання фільтрів 3x3 замість 11x11, використання нелінійної функції активації після кожної згортки. Таким чином, при зменшенні просторової інформації, функція класифікації стає більш дискримінаційною, і вся модель загалом показує кращі результати. Після створення даної моделі популярність глубинного навчання зробила ще один стрибок.

1.3.2 ResNet і залишкові блоки

Головна ідея побудови даної архітектури – використання залишкових блоків, приклад якого зображений на Рис. 1.2. З моменту публікації даної моделі у 2015, розробники створили багато її копій, таким чино покращивши час роботи, результативність, кількість шарів та інші параметри[8].

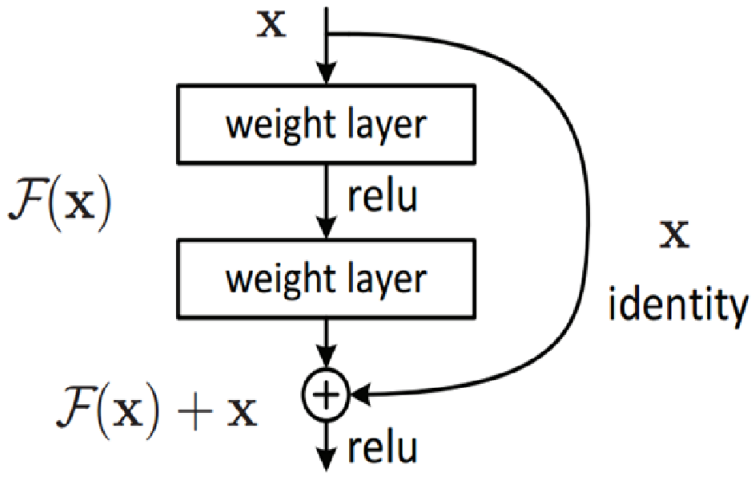

Рисунок 1.2 – Залишковий блок архітектури ResNet

Ця модель ще цікава тим, що першою обійшла результат людей у завданні класифікації. Ідея полягає в тому, що кожен наступний блок може брати потрібну йому інформацію з попередніх, і градієнти можуть поширюватись назад по мережі на порядок швидше ніж раніше. Це перша «настільки» глибока мережа,що містить в собі від 50 до 300 шарів, в залежності від варіанту реалізації.

1.3.3 DenseNet

Кожен шар даної архітектури з’єднаний з наступним в режимі подачі інформації. Завдяки цьому кожен шар може використовувати результати всіх попередніх, а власні мапи – як вхідні дані для наступного шару. Тут використовується вже не додавання,як в ResNet, а конкатенація. Дана мережа, на відміну від представлених вище допомагає зменшити проблему зникаючого

градієнту,одну з найважливіших в нейронних мережах. Також вона значно покращує поширення ознак, використовує повторне використання функцій, що допомагає зменшити кількість параметрів[9].

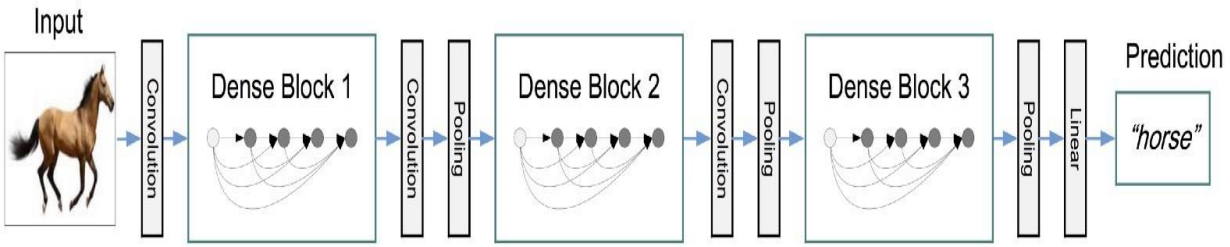

Рисунок 1.3 – Архітектура мережі DenseNet

Таким чином дана архітектура(Рисунок 1.3) була останньою, і показала один з найкращих результатів за весь час. Підсумувавши в собі прогрес за всі попередні роки в задачі класифікації зображень,вона продовжує конкурувати з новими моделями і цікавими рішеннями.

1.4 Загрози використання нейронних мереж

Разом з тим, як розвивалась архітектура сучасних нейронних мереж, з’явились і атаки на них. Поскільки класифікатори, що містять в собі такі моделі допомагають сильно зменшити затрати людських сил, а також автоматизують багато процесів, зокрема і для великих компаній та корпорацій. В зв’язку з широким використанням, вони містять в собі і великі ризики, при помилкових результатах класифікації. Система автентифікації може не пропустити того, кого мала б, або ж і зовсім, ідентифікувати зловмисника, як директора підприємства. Все це під силу певного роду атакам на алгоритми класифікації об’єктів[10].

Приклади атак на нейронні мережі – це такі вхідні дані до моделі машинного навчання, які атакуючий спеціально готує, щоб заставити модель зробити помилку. Це як оптична ілюзія, тільки для алгоритмів. В своїй роботі я зайнявся дослідженням саме таких вхідних даних, та методами захисту від подібних атак. Ця проблема дуже актуальна в наш час, атаки постійно розвиваються, і якогось одного рішення, для захисту від них не існує. Хоча серйозність проблеми надзвичайно висока.

Отже, атаки на нейронні мережі, поділяються на дві гілки: так звані “white-box” та “black-box” атаки. Якщо говрити про «чорний ящик», то в рамках атаки, це означає, що атакуючий не знає ні структури моделі, ні її параметрів ні виду мережі. В свою чергу «білий ящик» має на увазі, що атакуючий знає і вид моделі,і її гіперпараметри, і схему її побудови(кожен шар, їх розміри,і тд).

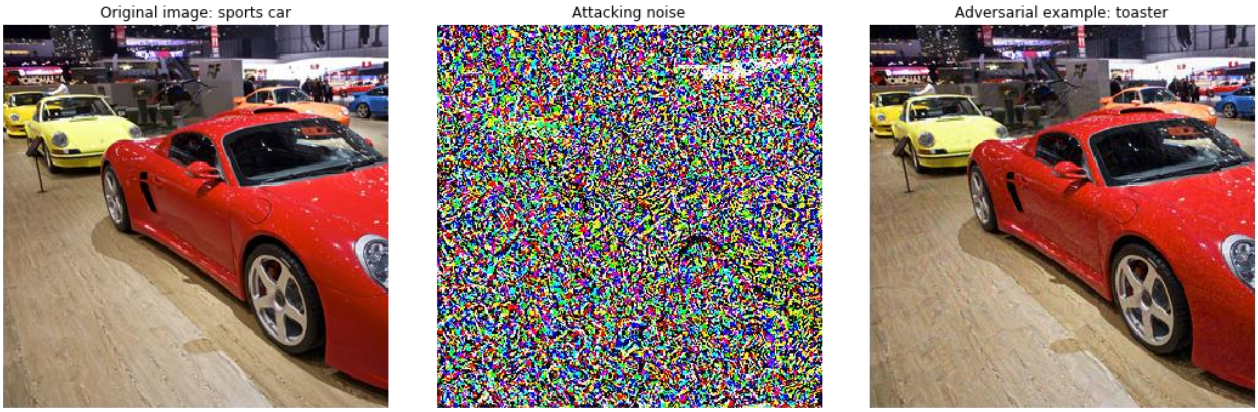

Рис 1.4 – Приклад атаки на нейронну мережу

Можемо бачити одну з найпростіших атак на класифікатор(Рисунок 1.4). На фото додається певного характеру шум,який майже не відрізняється для ока людини, та має сильний вплив на результат класифікатору. Як бачимо, першу картинку класифікувало як спорткар, а вже змінену нами – як тостер.

Таким чином, бачимо, що системи, які містять в собі глубокі нейронні мережі, фактично мають дуже високий ризик бути зламаними, і видавати невірні результати.

Висновки до розділу 1

У першому розділі нами були розглянуті системи класифікації зображень. Їх види, архітектури, призначення та застосування. Спочатку пояснили весь шлях, який має пройти зображення, щоб алгоритм відніс його до певної категорії, а потім порівняли методи які використовувались раніше,і поступово перейшли до більш сучасних архітектур.

Пояснили, чому нейронні мережі в наш час є важливим інструментом у багатьох завданнях комп'ютерного зору, а також вияснили, які саме їх архітектури використовуються для певних задач. Було проведено короткий огляд конкурсу з класифікації зображень ImageNet, де нейронні мережі домінують, починаючи з 2014 року. Було розглянуто кожну з найвідоміших і найпотужніших архітектур нейронних мереж,а саме VGGNet, ResNet, DenseNet. Вияснили, за рахунок яких змін вони змогли досягти таких серйозних результатів в завданні класифікації зображень.

Був проведений аналіз областей використання глубинних нейронних мереж в сучасності, та встановлено, що вони виконують важкі завдання, і навіть обходять людей у деяких функціях, і обов'язково потребують захисту. Для цього, оглянули основні загрози, що можуть вплинути на результати класифікаторів, та навели відповідні приклади, для того, щоб в подальшому спробувати адаптувати мережі до різних типів атак.

2 АТАКИ НА НЕЙРОННІ МЕРЕЖІ, ЩО ВИРІШУЮТЬ ЗАДАЧІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

2.1 Види атак

Атака на нейронну мережу – це такі вхідні зображення, що спричиняють хибний висновок моделі. Існує багато способів внести такі зміни в початкове зображення, щоб різко змінити роботу мережі. В цьому розділі ми розглянемо такі атаки як: випадкове перетворення, метод швидкого градієнтного спуску, стандартний ітеративний метод, атака одного пікселю, генерація зображень за допомогою змагальних нейронних мереж, просторово змінені приклади.

2.1.1 Метод випадкового перетворення

Найкраще буде показати роботу даної атаки на прикладі, а потім пояснити її дію.

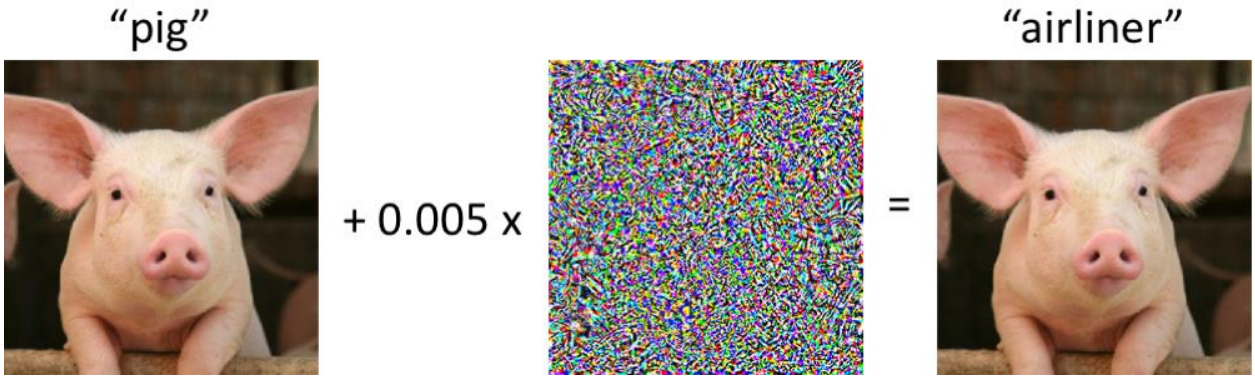

Рисунок 2.1 – Приклад роботи додавання випадкового шуму

На Рис.2.1 видно результат роботи атаки на одну з найпопулярніших архітектур для класифікації зображень – ResNet50. Зліва бачимо правильний висновок класифікатору – клас «свиня». Після перетворення зображення, пікселі якого були змінені не більше ніж на 0.005, результат роботи мережі тепер клас «авіалайнер». При чому, модель впевнена,що це саме клас

«авіалайнер» з великою точністю(близько 95%). Такі атаки з’явились ще в 2004 році, і в 2006 році вже працювали над генеруванням прикладів, для погіршення результатів нейронних мереж, що займаються задачами класифікації зображень. Поступово таким атаками стали приділяти все більше уваги, починаючи з 2013 року, коли вияснили, що сучасні нейронні мережі ще більш до них вразливі. Загалом даний метод один з найпростіших, і включає в себе просто перебір різного типу шумів, помножених на якийсь, порівняно невеликий параметр, та додавання такого шару на зображення. Після додавання, для людського погляду майже неможливо відрізнити початкове і перетворене зображення, а для нейронної мережі це справжній виклик, що потребує додаткових затрат на тренування. Мінус методу в тому, що повний перебір параметрів та шумів займає дуже багато часу, і обчислювальних потужностей. Тому на зміну йому прийшов новий метод, що дозволяє з кожною ітерацією рухатись в напрямку погіршення результатів мережі.

2.1.2 Метод швидкого градієнтного спуску

Один з найголовніших алгоритмів генерації атак на нейронні мережі. Нехай x це оригінальне зображення, що подається на вхід мережі. Тоді за y позначимо клас, до якого має належати об’єкт, що на ньому зображений. Тоді θ – ваги мережі, на яку збираємось проводити атаку. І позначимо виразом $L(\theta, x, y)$ – як функцію помилки, що використовується для тренуваннямережі. Спочатку ми рахуємо градієнт для функції помилки, зважаючи на вхідні пікселі.

$$\nabla_x L(\theta, x, y) \qquad (2.1)$$

Де ∇- оператор, що означає апроксимований алгоритм, що дозволяє брати похідні функції по багатьох її параметрах(2.1). Можна уявити це як матрицю розмірами (висота, ширина, кількість каналів), що містить в собі певні параметри. Як і раніше, нас цікавить лише знак нахилу, щоб знати в яку сторону рухатись, а саме дізнатись збільшувати, чи зменшувати нам значення

тих чи інших пікселів[11]. Ми множимо цей знак на дуже маленьке значення ε, щоб впевнитись в тому, що ми робимо не надто великий крок, і рухаємось лише в правильному напрямку. Це і буде нашим перетворенням.

$$\eta = \varepsilon \, \mathrm{sign}(\nabla_x L(\theta, x, y)) \qquad (2.2)$$

Де η – крок, який ми маємо зробити після однієї ітерації градієнтного спуску. Таким чином наше фінальне зображення, це вхідне + перетворення, які ми згенерували завдяки градієнтному спуску.

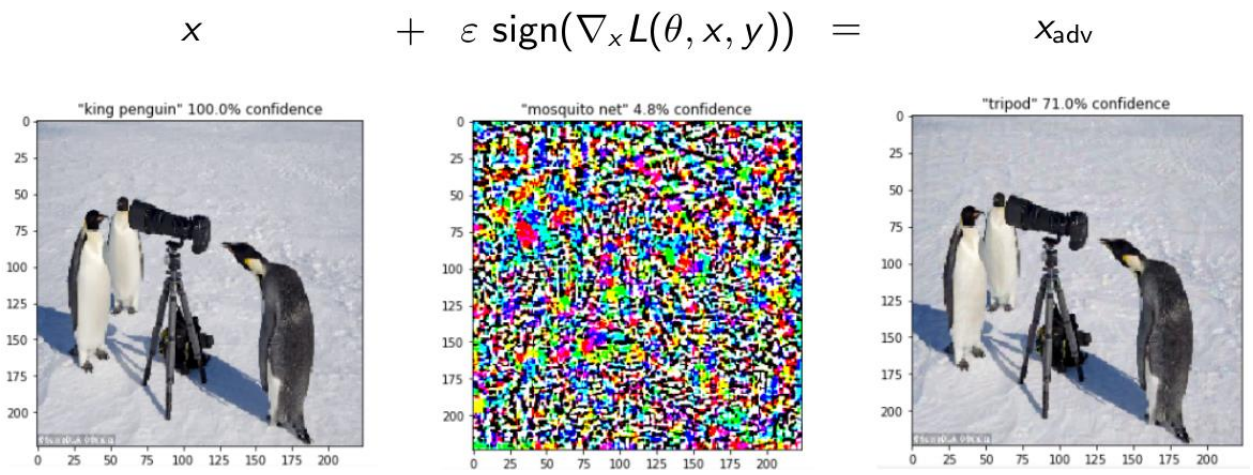

Рисунок 2.2 – Практична реалізація атаки,що основана на швидкому градієнтому спуску

Даний метод використовує структуру нейронної мережі, її параметри та результати, тому алгоритм атаки віднесемо до атак групи «білий ящик».

2.1.3 Атака одного пікселю

Назва даної атаки говорить сама за себе. Зловмисник намагається суттєво змінити висновок нейронної мережі зміною всього одного пікселю. Дане перетворення пожна здіюснювати у декілька способів. Найперший та

найпростіший метод – пошук пікселю випадковим вибором. Ітеративно,ми змінюємо якийсь піксель на зображенні на певне значення, і подаємо на вхід моделі. Такі ітерації відбуваються доки ми не спробуємо всі можливі варіанти, або доки ми не отримаємо потрібний нам результат(а саме хибну класифікацію об’єкту до певної категорії)[12].

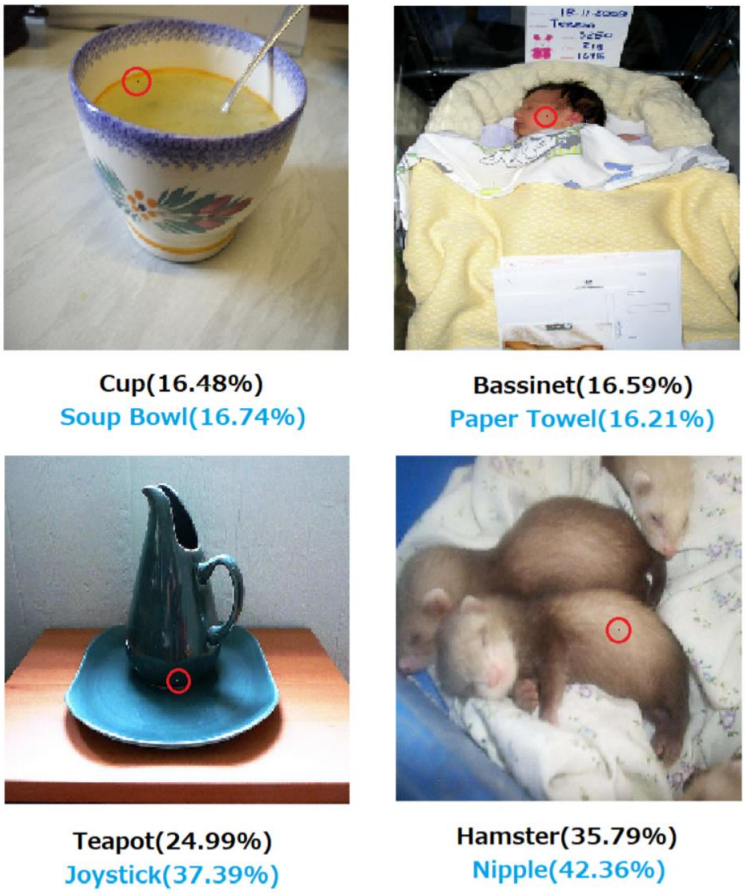

Рисунок 2.3 – Приклад атаки одного пікселю

Приклади виконання атаки заміни випадкового пікселю на зображенні, і результати роботи мережі на таких об’єктах(Рисунок 2.3) показує, що така атака досить проста у виконанні, але вона має декілька мінусів. Перший – ми не можемо замінити висновок нейронної мережі на потрібний нам, лише на випадковий, якщо підберемо правильний піксель. Другий – атака працює не на

всіх зображеннях, деякі зображення не вийде змінити будь-якими замінами одного пікселю, цього для зміни результату буде недостатньо.

Для того щоб все ж знайти потрібний там піксель, і його значення, науковці розробили метод, що бере за основу «диференціальну еволюцію».

Він потребує менше інформації про середовище, і є атакою «чорного ящику», і може спрацювати на більшій кількості нейронних мереж, в зв'язку з властивими їм особливостями диференціальної еволюції. Результати такої атаки показують, що 68 % з набору даних CIFAR-10, та 16 % вже згадуваного раніше набору даних ImageNet можуть бути перетворені, хоча б на одну категорію, відмінну від вірної. При чому відбувається така атака з точністю в 73% та 23% відповідно, для кожного набору даних. Таким чином даний спосіб атаки показує, що існують алгоритми, спроможні зрозуміти, яким саме чином нейронна мережа робить свій висновок, і внести зміни в найпотрібнішу частину зображення. Вона найбільш сильно показує, що глибинні нейронні мережі є вразливими до подібних атак, на зображеннях з невеликим розширенням.

2.1.4 Генерація зображень за допомогою змагальних нейронних мереж.

Існує багато моделей, що мають за основу змагальні нейронні мережі, але в цьому пункті мною була розглянута одна,а саме AdvGAN[13]. Трохи краще з тим як працюють моделі такого типу ми розглянемо в наступному пункті, коли будемо описувати механізми захисту від атак, але поскільки атаки створені завдяки генеративним моделям одній з найнебезпечніших, то зупинимось на них і тут.

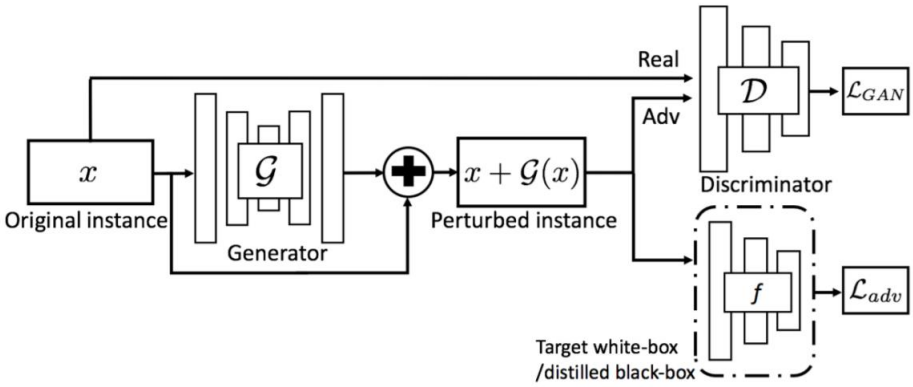

Рисунок 2.4 – Архітектура змагальної нейронної мережі, для проведення атак

Структура моделі містить в собі дві нейронних мережі – генератор та дискримінатор. Як дискримінатор обрано моделі, які використовуються у сучасних системах комп’ютерного зору. Тому загальна схема роботи даного методу така: генератор намагається створити зображення, яке буде максимально подібним до об’єкту певної категорії, але дискримінатор буде давати висновок, що це об’єкт іншої, потрібної нам категорії. Таким чином, одночасно, зображення проходить через дві моделі – одна навчається створювати зображення, які будуть використані для атаки, а інша,з замороженими вагами, каже нам, до якого класу відноситься той чи інший об’єкт. Ця атака є найважчою в реалізації, але одночасно найпотужнішим механізмом злому навіть найглибших нейронних мереж, захищених від атак.

Найбільший плюс такої архітектури - ми можемо самі обирати до якого класу буде належати наше вхідне зображення.

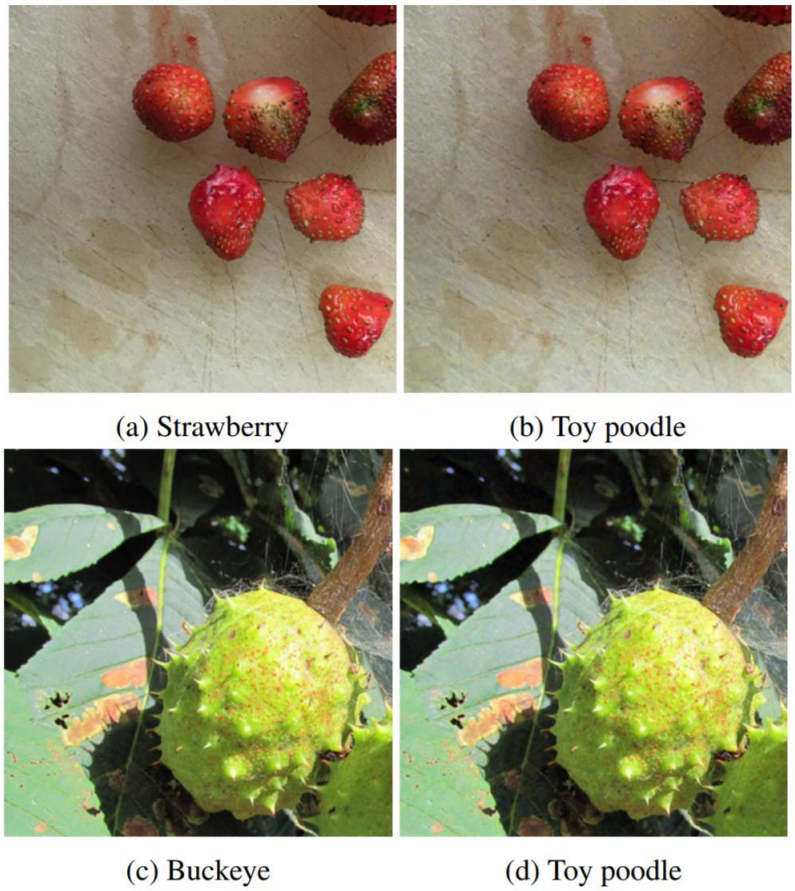

Рисунок 2.5 – Результат роботи атаки побудованої на змагальних мережах

Результати добре натренованої генеративної моделі(Рис. 2.5), показують, що ми можемо змінити результати класифікатору на потрібний нам, не сильно змінюючи зображення для людського сприйняття. Класи «полуниця» та «каштан» перетворились на клас «іграшковий пудель», хоча були внесені мінімальні зміни, завдяки роботі AdvGAN.

2.1.5 Атака Карліні-Вагнера.

Розглянемо одну з найпотужніших атак, що з’являлись у даній сфері за останні роки,а саме – атаку Карліні-Вагнера[14]. Ця атака спроможна обійти такі сильні алгоритми захисту як дистиляція та змагальне тренування. За

допомогою даного підходу можна знизити точність нейронної мережі з класифікації більше ніж на 90%, і є найнебезпечнішою із всіх існуючих атак.

Звернемося до основного підходу для побудови атакованих прикладів. Для початку звернемося до початкового формулювання проблеми, що формально визначає проблему знаходження змагального прикладу для зображення x наступним чином:

$$\min D(x, x + \delta) \qquad (2.3)$$

де x – фіксоване зображення, і наша ціль знайти таку δ, що мінімізує функцію (2.3). Ось чому, ми намагаємось знайти найменше значення δ, що спричинить помилку у класифікації нейронною мережею, але зображення буде залишатись таким, як і раніше, для людського погляду. В даному випадку D це одна з метрик, наприклад L_0, L_2, or $L\infty$. Така проблема вирішується, формулюючи її, як відповідну задачу оптимізації, і яка,власне, вирішується, вже за допомогою існуючих алгоритмів оптимізації. Є багато способів для таких дій: автори даної атаки досліджують простір формулювань, і емпірично визначають, який з них призводить до найбільш ефективних атак.

Вирішення вищепоставленої задачі надто громіздке для існуючих алгоритмів оптимізації, поскільки функція (2.4) надто нелінійна.

$$C(x + \delta) = t \qquad (2.4)$$

Для полегшеня задачі використовується функція *f*, така що $f(x + \delta) < 0$. Є багато варіантів вибору для такої функції, але це не стосується даної роботи, і вимагає ще багато досліджень. Але ми маємо знати, що тепер задача оптимізації з такої (1), зводиться до такого виду:

$$\min D(x, x + \delta) + c \cdot f(x + \delta) \qquad (2.5)$$

де c – параметр, який ми маємо підібрати, і який відіграє ключову роль у проведенні атаки. Пропонується використовувати для оптимізації три найбільш ефективних методи:

- Проективний градієнтний спуск
- Частковий градієнтний спуск
- Заміну зміних(в даному випадку заміняємо δ на ще більш нелінійну конструкцію, і розглядати даний метод як зглажений частковий градієнтний спуск)

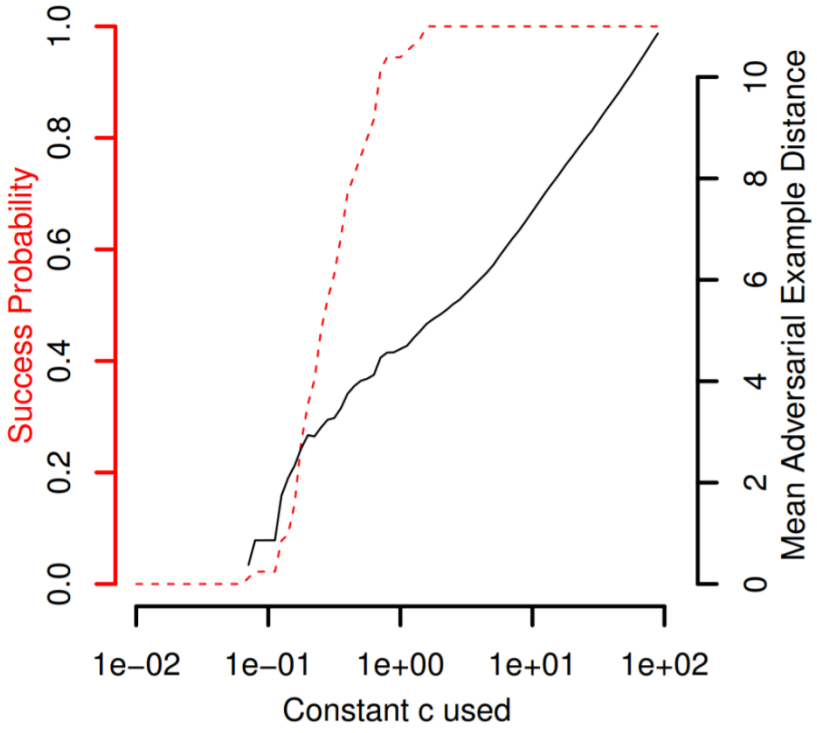

Рисунок 2.6 – Чутливість вибору константи c

На рисунку показано ефективність роботи нашої атаки, в залежності від вибору константи c. Бачимо, що у випадку коли c < 0, атаки рідко бувають успішними. Після того як ми збільшуємо c до значення, що перевищує одиницю, атака стає менше ефективною, але завжди успішною. Емпірично було встановлено, що найкращий шлях для вибору c – використовувати найменше значення, таке що задовільняє умову $f(x^*) \leq 0$, де x* - зображення, на яке вже проведена атака. Такий вибір призводить до того, що градієнтний

спуск мінімізує обидві змінних одночасно, замість вибору,оптимізувати один чи інший. У даному алгоритмі також присутні такі зміні як кількість кроків градієнтного спуску, та кількість перезапусків роботи оптимізатору, для покращення результатів роботи.

Отже, як вже було сказано, використовуючи дану атаку, добре підібравши параметр c, та правильно сформулювавши задачу оптимізації, можемо отримати результат, що справді дивує і заставляє задуматись, про використання нейронних мереж як таких.

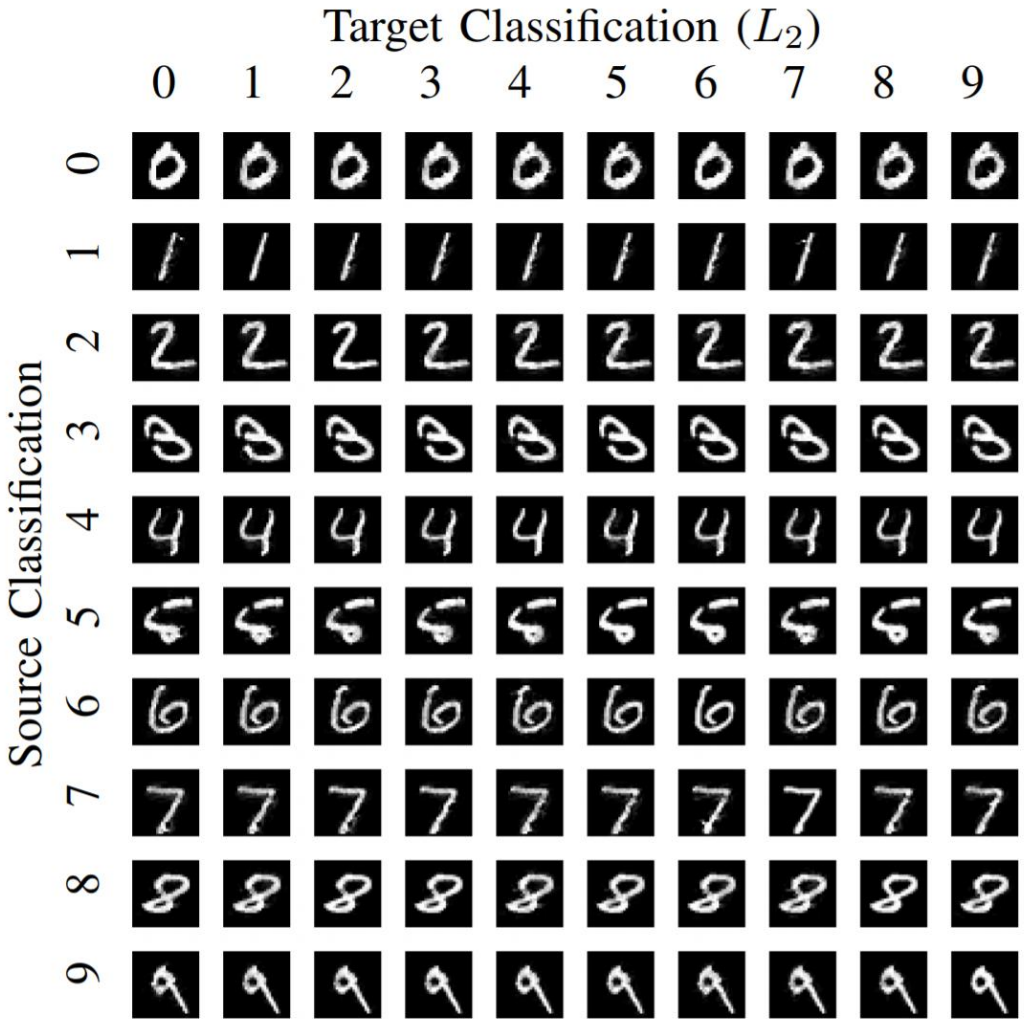

Рисунок 2.7 – Результати атаки Карліні-Вагнера, з метрикою L_2

На зображенні (Рис. 2.7) можемо бачити результати роботи методу. Тобто ми можемо змінити будь-яку цифру на будь-яку іншу, без помітних артефактів на самому зображенні. Не варто говорити про те, як часто нейронним мережам доводиться розпізнавати цифри, і як цим можна скористатись(починаючи від цін та номерів автомобілів, закінчуючи банківськими рахунками). Якщо змінити цифру на іншу для цієї атаки не проблема, то можемо продемонструвати(Рис. 2.8 та Рис. 2.9), як змінюють просто чорний та білий квадрат, для різних метрик, на таку цифру, яку ми самі захочемо.

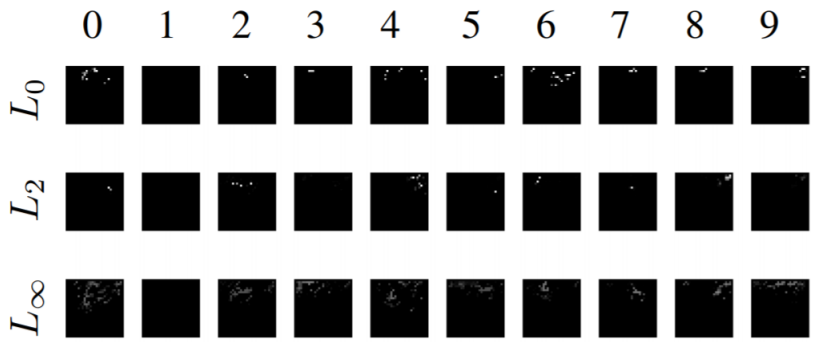

Рисунок 2.8 – Зміна результатів класифікації нейронної мережі з чорного квадрату, до цифри, яку ми захочемо бачити в результатах

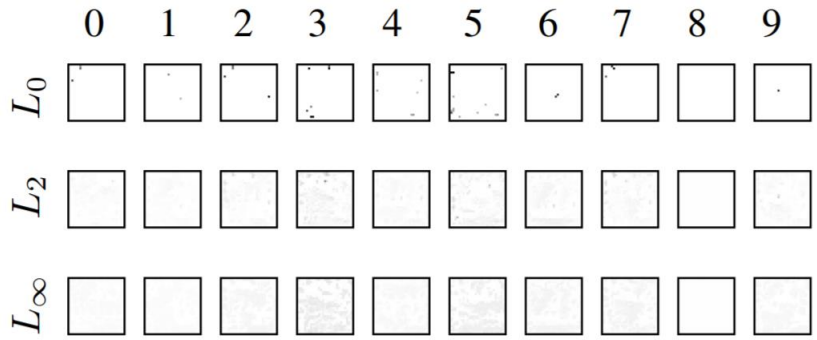

Рисунок 2.9 - Зміна результатів класифікації нейронної мережі з чорного квадрату, до цифри, яку ми захочемо бачити в результатах

2.2 Огляд методів захисту від атак на нейронні мережі класифікації зображень

Існує багато підходів, для захисту глубинних нейронних мереж від атак. Але головна їх проблема в тому, що кожен елемент захисту намагається адаптувати мережу до одного виду атаки. В загальному, не існує ніякого прийнятого всіма рішення, для захисту від зловмисних атак. Багато методів захисту були запропоновані зі змішанними результатами, і часто упускаються науковцями в подальших дослідженнях. Далі буде наведено декілька найвживаніших і дієвих прикладів:

- Змагальне тренування – використання вже змінених зображень(тобто зображень на які вже була проведена атака) під час тренування нейронної мережі, задля зменшення помилки при класифікації. Для формулювання такої концепції, та покращення навчання використовують різні методи. Раніше нами вже були розглянуті види атак, тому ми не будемо вдаватися в тонкощі реалізації прикладів для тренування мережі. Але в загальному план для захисту моделі такий: беремо наш вхідний набір даних, проводимо над кожним зображенням ряд перетворень, намагаючись погіршити результати роботи мережі, і додаємо ці зображення до тренувального набору даних. В цілому змагальне тренування дало неоднозначні результати, але все ж робить модель стійкішою до різних типів атак.
- Автоенкодери. Автоенкодери це один з типів глубинних нейронних мереж, який спочатку зменшує розмірність вхідного зображення, пропускаючи його через нижні шари, перед тим як відновити інформацію у вихідному шарі, що буде мати таку ж розмірність як і початкове зображення. Іншими словами, він намагається стати тотожною функцією, але крім того, що спочатку

зжимає зображення, а потім намагається відновити її. Така архітектура має нівелювати шум, якщо він присутній у вхідному зображенні, і зберігає лише такі образи та функції, які необхідні для відновлення початкового об’єкту. Такі автоенкодери також допомагають прибрати втручання у вхідне зображення зловмисників, що проводили атаку на нейронну мережу, і прибрати деякі елементи цієї атаки. Проблема методу в тому, що вона залишається вразливою до атак «білого ящику», нічого не заважає нам внести такі зміни, які будуть відновлюватись навіть автоенкодером.

- Захисна очистка. Захисна очистка це ще один шлях вирішення проблеми помилкових результатів нейронної мережі, під впливом атак. Даний метод захисту пропонує використання двох нейронних мереж замість однієї. Весь шлях вхідного зображення виглядає так – ми отримуємо на вхід зображення, воно проходить через всю архітектуру першої нейронної мережі, але замість вибору максимального значення з вектору ймовірностей, ми зберігаємо сам вектор в пам’яті. Інша нейронна мережа приймає на вхід вже не зображення, а вектор ймовірностей розподілу першої моделі. Таким чином вся наша архітектура стає більш гладкою, та менш чутливою до змін у вхідному зображенні, а значить, адаптованою де певних типів атак.
- DeepSafe. Один з останніх найперспективніших засобів в боротьбі з атаками. Даний метод заснований на факті, що всі входи що належать області вхідного простору, належать до одного класу, і мають бути позначені однаково. Тобто суть алгоритму заключається в тому, що для кожного класу завжди існує одне зображення правильно класифіковане, і якимось чином розміщене у просторі(сучасні нейронні мережі дозволяють перетворити

вхідне зображення у вектор будь-якої розмірності,частіше всього використовують 1024). Коли ж на вхід мережі подається нове зображення, воно таким же чином перетворюється у вектор, і розміщується у просторі. Далі до нього шукається найближче по косинусній відстані(або будь-яким іншим відомим методом), і порівнюються висновки останнього шару нейронної мережі. Якщо два вектори знаходяться далеко один від одного, але нейронна мережа присвоїла їм однакову категорію – це сигнал, що на зображення була проведена атака, і можливо варто перейти до більш точних алгоритмів захисту. Мінус даного методу у швидкості виконання – перед тим як зробити висновок, ми маємо ще отримати вектор, і порівняти його з іншими. Але з розвитком обчислювальних потужностей, і модернізацією самого методу, він може стати одним з найкращих в своїй області.

Після загального огляду більшості існуючих методів захисту від атак на нейронні мережі, ми плавно переходимо до основного об’єкту мого дослідження, а саме – змагальні нейронні мережі. А поскільки останні декілька років саме дана архітектура нейронних мереж розвивається більше всього, то і алгоритми захисту побудовані на них мають успіх у вирішенні поставленої задачі.

2.3 Змагальні нейронні мережі, та архітектури що на них побудовані

Змагальні нейронні мережі встигли отримати титул «найцікавіша ідея за останні 10 років у машинному навчанні». Все почалося з того, що Ian J. Goodfellow у 2014 році представив статтю з назвою «Генеративні змагальні моделі», що створило справжній фурор у сфері глубинних нейронних мереж, і машинного навчання в цілому[15].

Змагальні нейронні мережі відносяться до генеративного типу. Це означає, що вони здатні виготовляти\генерувати якусь нову інформацію, а не лише оцінювати вже наявну. Звичайно, що ця здатність, створювати асболютно нові об’єкти спочатку викликає неоднозначні емоції з приводу того, як працює дана архітектура. В загальному, змагальна нейронна мережа має у своїй структурі два обов’язкових пункти – модель-генератор та модель-дискримінатор. Щоб показати різницю між дискримінатором та генератором, наведемо простий приклад: якщо у нас є речення, яке ми хотіли б класифікувати, тобто отримати якусь його оцінку(позитивне чи негативне, до якого класу відноситься і т.д.) ми б подали його дискримінативній моделі, яка проводить оцінку отриманих вхідних даних, і на виході отримали б певний клас об’єкту. Тобто дискримінативні моделі проводять певний маппінг між вхідними даними і назвами класів. Вони направлені виключно на знаходження цієї кореляції.

Один із способів опису генеративних моделей – це повна протилежність дискримінативним. Вони намагаються з отриманої назви класу згенерувати реальні дані. Ще один шлях визначити різницю між такими моделями полягає в тому, що:

- Дискримінативні моделі намагаються вивчити, і провести у просторі границю між класами.
- Генеративні моделі вивчають розподіл даних у межах одного класу.

Отже, якщо коротко описати роботу змагальних нейронних мереж, то суть полягає в тому,що одна мережа(що зветься модель-генератор) створює нові дані, а інша(модель-дискримінатор) вказує, наскільки схожі дані згенеровані першою моделлю до реальних вхідних даних. Тобто дискримінатор вирішує задачу бінарної класифікації, і вказує, відноситься згенероване зображення до набору тренувальних даних,чи ні. Обидві нейронні

мережі тренуються одночасно. Описати один крок роботи моделі можна таким чином:

- Генератор приймає на вхід випадкові числа, і зважаючи на них генерує певне зображення.
- Це згенероване зображення «скормлюється» на вхід моделі-дискримінатору, одночасно з зображеннями, які взяті з реального набору даних, приготованого заздалегідь.
- Дискримінатор приймає обидві картинки, і робить свій висновок. А саме повертає вектор ймовірностей, кожне число якого знормоване до проміжку [0,1], і представляє з себе ймовірність того, що зображення відноситься до реальних вхідних даних.

Отже ми маємо два зворотніх зв’зки – дискримінатор з’єднаний з реальним вектором розподілу ймовірностей, який заданий нашим набором даних, а генератор – пов’язаний петльою зворотнього зв’язку з дискримінатором[16].

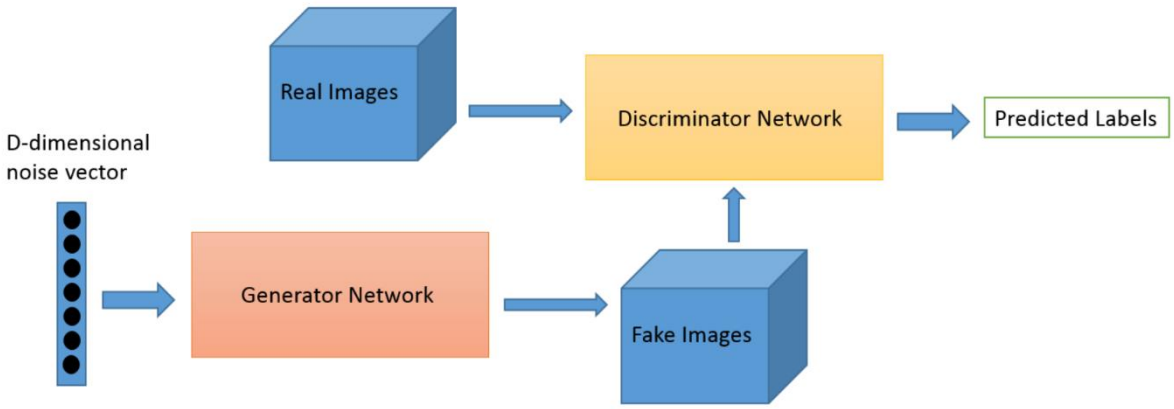

Рисунок 2.10 – Схема роботи найпростішої змагальної нейронної мережі

Якщо говорити про найпростішу архітектуру таких моделей(Рисунок 2.10), то в сучасних алгоритмах використовується їх сильно модифікована

копія. Одну з таких модифікованих версій було вирішено використати, для захисту нейронних мереж від атак, на прикладі класифікації зображень, і її назва – DefenseGAN.

Висновки до розділу 2

У другому розділі було розглянуто теоретичну частину, для виконання дипломної роботи. В першу чергу ознайомились зі всіма можливими атаками, на найронні мережі, що вирішують задачі класифікації зображень. А саме: атака завдяки додаванню випадкового шуму, додавання спеціального виду шуму за допомогою методу швидкого градієнтного спуску, атака одного пікселю, та атака, побудована на змагальних нейронних мережах. Для кожного виду атак були наведені приклади, та пояснення, як побудувати таку атаку на будь-якому вхідному зображенні.

Інша частина розділу присвячена сучасним методам захисту. Був проведений короткий огляд кожного з методів, та наведений приклад його роботи. Кожен підрозділ присвячений одному з методів захисту від атак на нейронні мережі : змагальне тренування, захисна очистка, DeepSafe та DefenseGAN. Також повністю описали роботу змагальних нейронних мереж, їх типову архітектуру та методи їх модернізації.

Таким чином, розділ повністю присвячений теоретичній частині, необхідній для подальшої практичної реалізації потрібних нам алгоритмів.

3 ПОБУДОВА МЕТОДУ ЗАХИСТУ

3.1 Побудова архітектури нейронної мережі, що буде використовуватись в якості захисного механізму

За останні роки було запропоновано багато способів захисту від змагальних атак на нейронні мережі. Ці алгоритми захисту можна поділити на три основних групи : зміна тренувальних даних, для того щоб зробити класифікатор надійнішим і стійкішим до атак(наприклад збільшення даних для тренування, за рахунок додавання до них вже атакованих зображень), підготовка класифікатору, для зменшення кроку градієнту(для прикладу – захисна дистиляція, про яку було сказано раніше), і спроби звести до мінімуму шум, створений зловмисником, перед тим як подавати такі фото на вхід. Всі ці підходи мають певні обмеження - вони зачасту ефективні лише проти атак «чорного ящика» або атак «білого ящика», але ніколи одночасно. Крім цього мінусу, є ще один, не менш важливий. Деякі з вищеперечислених алгоритмів розроблені з урахуванням атак на якісь певні моделі, і не надто ефективні проти нових, модернізованих атак.

Тому в подальшому буде розглянуто рішення, яке ефективне і проти атак «білого ящику» і проти атак «чорного ящику». Пропонується використовувати за основу такого алгоритму змагальні нейронні мережі, щоб зменшити вплив змагальних перетворень, та робити проекцію вхідних даних на діапазон моделі-генератора, перед тим як подавати дані на класифікатор[17]. У рамках даної архітектури дві моделі навчаються разом, і обидві мають не заморожені ваги, які постійно змінюються: модель-генератор намагається повторити розподіл вхідних даних, а модель-дискримінатор робить висновок, реальні дані прийшли їй на вхід, чи ті, що були виходом генератору. В даному випадку модель-генератор вивчає відображення G з ветктору малої розмірності $z \in R^k$, в високого розширення зображення в просторі R^n. Архітектура нашої мережі передбачає такий метод захисту, як зняття шумів з вже атакованого

зображення,перед тим як подавати його на вхід мережі. Ідея полягає в тому, що перед тим як подавати вхідні дані, ми проектуємо їх на простір моделі-генератору, і намагаємось мінімізувати помилку || G(z) – x||^2, використовуючи для цього кроки градієнтного спуску. Вже отримане з моделі фото подається на класифікатор. Поскільки модель-генаратор тренується на незашумлених зображеннях, то вона намагається зпроектувати такі ж зображення на виході, і очікується, що доданий крок допоможе сильно зменшити будь-якого можливого шуму.

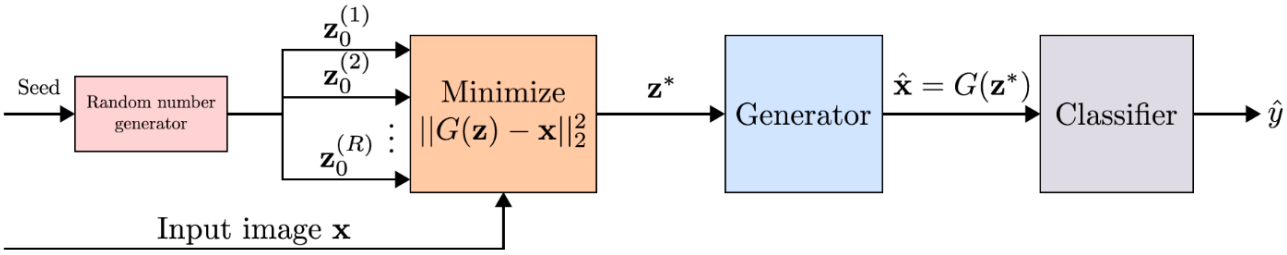

Рисунок 3.1 – Загальна архітектури DefenseGan

На перший погляд схема мережі(3.1) не надто відрізняється від звичайної архітектури змагальних мереж. Але далі буде розглянуто такі моменти як спеціальні функції втрат, і особливості обробки зображень до того як подавати їх на вхід мережі.

Мережа тренується на будь-якому наборі даних, максимально можливої величини, навчання без вчителя(тобто нам потрібні лише зображення, без категорій, до яких вони відносяться). Класифікатор може бути навчений на початковому наборі даних, видозмінених різними атаками, зашумлених,або ж суміші даних наборів. У порівнняні з простою схемою змагальних нейронних мереж, і існуючими алгоритмами захисту, даний метод відрізняється декількома аспектами:

- Дана архітектура може використовуватись з будь-яким класифікатором, не змінюючи ні його структуру, ні ваги. Тому його можна розглядати як

етап попередньої обробки зображення, або ж просте доповнення до основної архітектури.

- DefenseGAN може бути використаний для захисту від будь-якої атаки. Він не намагається підлаштуватись під певну модель атаки, а може реагувати на всі типи атак, завдяки репрезантивній потужності змагальних мереж.
- Архітектура мережі сильно нелінійна, тому атаки «білого ящику», що основані на методі градієнтного спуску буде важко реалізувати.
- Якщо змагальна нейронна мережа досить репрезентативна(тренувалась достатню кількість часу, на великому наборі даних), то перенавчання її для кожного окремого завдання не є необхідним, і будь-яке зниження продуктивності, після додавання цієї мережі до основної не буде значним.

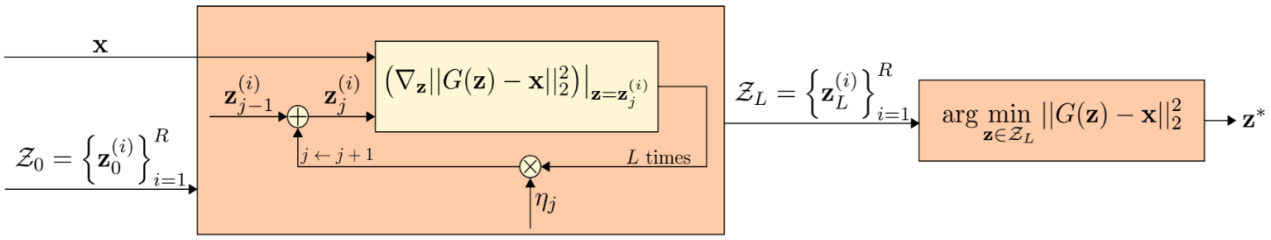

Рисунок 3.2 – Структура генератору DefenseGAN

Схема(Рис. 3.2) демонструє, що кожен етап спуску градієнту використовується нами для оцінки проекції зображення на діапазон генератору.

3.2 Постановка задачі для експериментів

Далі в роботі будуть проводитись експерименти, які показують фактичну стійкість DefenseGAN до атак різного рівня та типу:

1. Атаки «чорного ящику» - зловмисник не має доступу до деталей роботи класифікатору, структури, параметрів і стратегії захисту. Тобто він тренує свою нейронну мережу, щоб перевіряти на ній створені атаки.
2. Атаки «білого ящику» - зловмисник знає всі тонкощі реалізації мережі, на яку збирається проводити атаку. Він може рахувати градієнти, перебирати параметри, змінювати швидкість навчання. Тим більше, він має доступ не лише до класифікатору, а і до мережі, що відповідає за його захист.
3. Атаки «білого ящику» , як і в попередньому пункті. Тільки тепер, береться за увагу, що зловмисник ще й має доступ до,в нашому випадку, генератору випадкових чисел та послідовностей, з якого починається робота DefenseGAN. Тобто, фактично всі випадкові ініціалізації вектору $\{z_0^{(i)}\}_{i=1}^{R}$.

Результати роботи мережі, будуть порівняні з результатами змагального тренування, та MagNet при таких атаках як випадкове зашумлення, метод швидкого градієнтого спуску, та суміші цих двох атак. Мережі для атаки вибирались випадково, у них різна кількість навчальних параметрів, різна глубина, кількість шарів, та різні функції активації. Всі експерименти було проведено на сервері з процесором Intel Core i7-9700K 3.6GHz/8GT та відеокартою NVIDIA GeForce GTX 1080TI. Набори даних, які було використано в роботі це MNIST(набір рукописних цифр), та набір даних CelebA(набір фотографій знаменитостей з вказаними класами(нас цікавить лише гендер, як бінарна класифікація)). MNIST містить в собі 60000 зображень для тренування, та 10000 для тестування.

3.3 Результати навчання змагальної нейронної мережі

Для того, щоб забезпечити класифікатору захист від атак, ми спочатку маємо натренувати змагальну нейронну мережу генерувати зображення, що

схожі до тих, які подаються на вхід класифікатору. Отже, ми натренували дві моделі, одну на наборі даних з рукописними цифрами, іншу на наборі даних з знаменитостями. Будемо вирішувати дві задачі класифікації – бінарну та мультикласову(кількість класів > 2, а в нашому випадку – 10).

Рисунок 3.3 – Приклад роботи мержі, що генерує обличчя

На зображенні(Рис 3.3) можемо бачити результат однієї з ітерацій змагальної нейронної мережі, що навчалась на обличчях знаменитостей. Головна її задача – зрозуміти як виглядають обличчя обох гендерів, та вміти як генеруваи їх, так і відрізняти генеровані від справжніх.

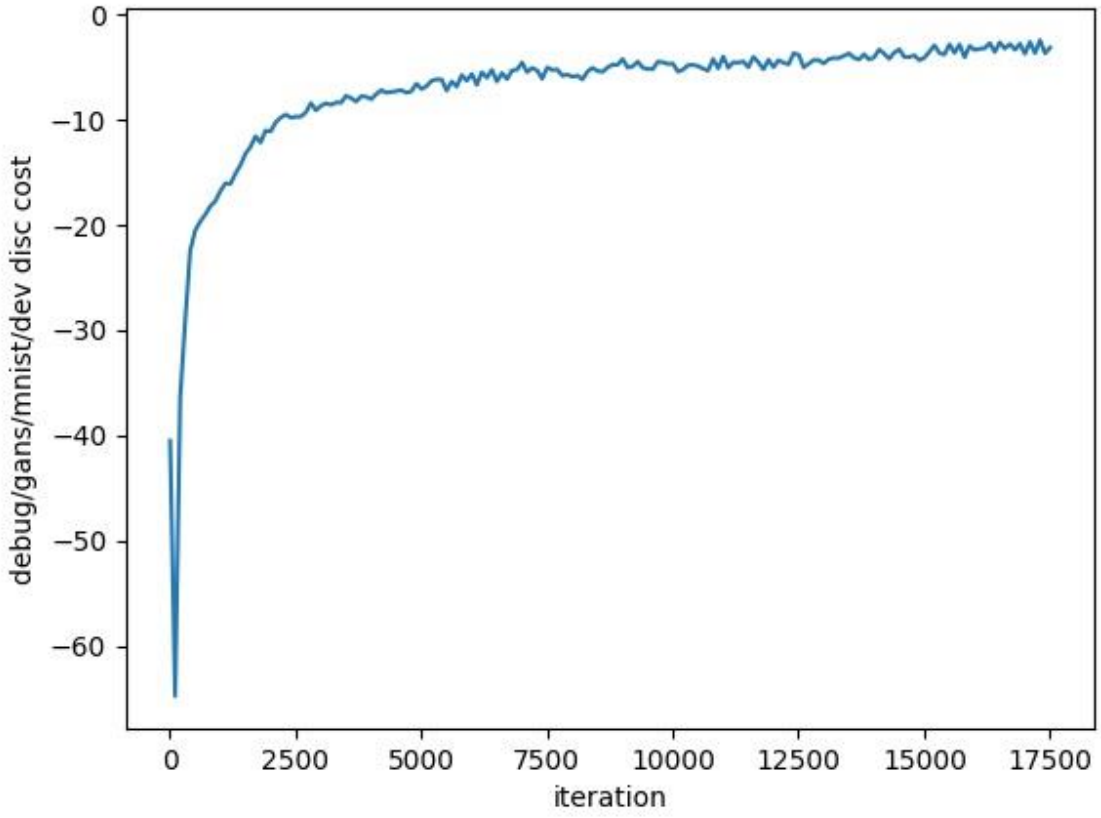

Рисунок 3.4 – Результати тренування мережі(набір даних для тренування)

Хотілося б зауважити, що метрика задана таким чином, що 0 досягається лише при ідеальній роботі мережі, тому тренування було зупинено приблизно на значенні -4.3, але цього цілком досить для того, щоб показувати хороші результати на завданні генерування облич.

Тепер перевіримо результати на тестовому наборі даних, задля того щоб переконатись, що робота мережі генералізована, і не заточена під тренувальні данні.

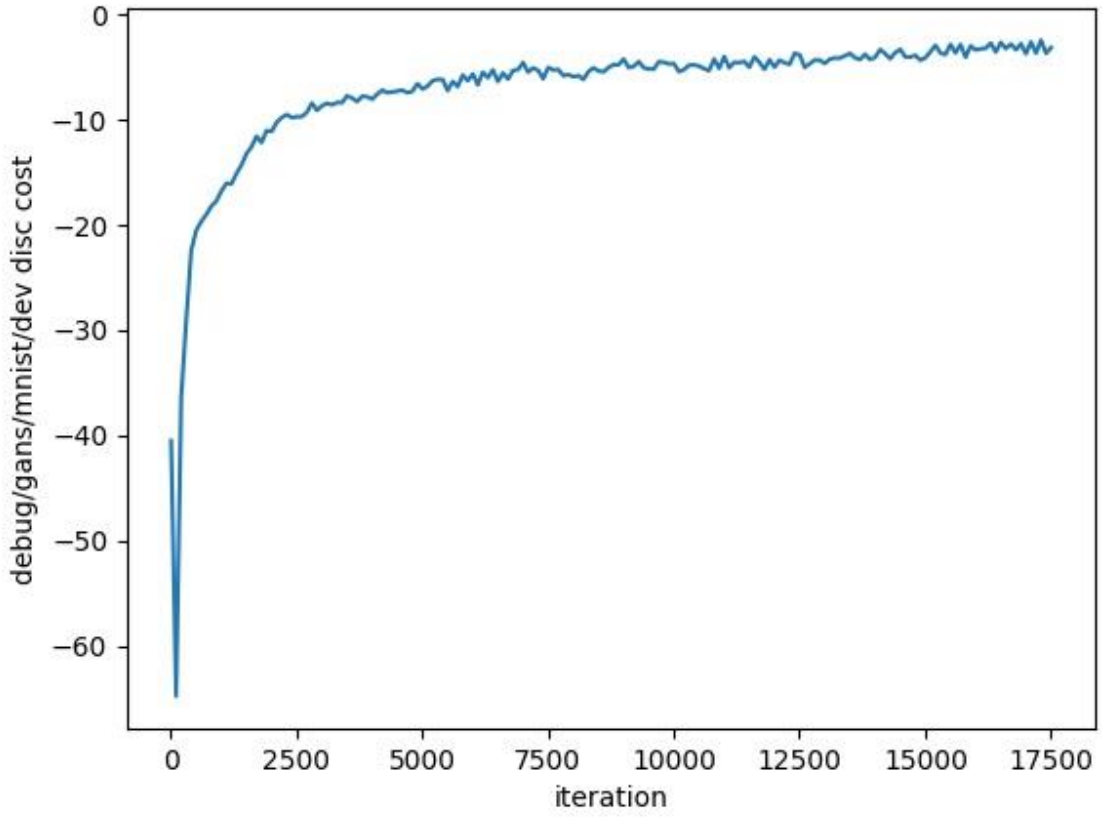

Рисунок 3.5 – Результати тренування мережі(набір даних для тестування та валідації)

Після того, як модель пройшла певну кількість ітерацій,можемо спостерігати за її точністю на тестовому та тренувальному датасеті.(Рис 3.4 та Рис.3.5 відповідно). Бачимо, що на графіку валідаційного набору даних(Рис 3.5) результати не сильно розходяться з тренувальним набором, тому можемо зробити висновок, що нейронна мережа добре генералізувала поданий набір даних, і як генератор, так і дискримінатор працюють добре. Отже модель готова для використання в якості алгоритму захисту. Перейдемо до другого набору даних, що містить в собі десять класів рукописних цифр.

З такими ж параметрами, як і попередній набір даних зі знаменитостями, було натреновано нову модель. Поскільки генерувати цифри набагато

простіше завдання для моделі-генератору, ніж обличчя людей, мережа тренувалась набагато швидше, і з меншою кількістю ітерацій.

Рисунок 3.6 – Результат генеративної моделі на наборі даних з цифрами

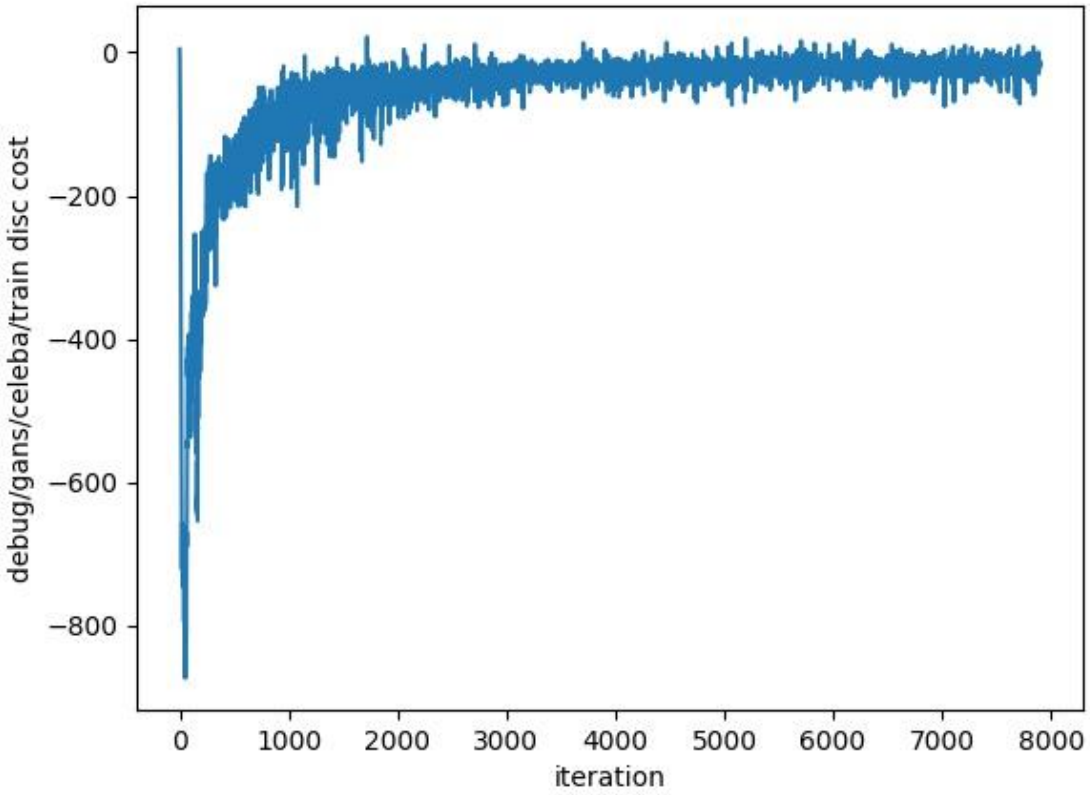

Рисунок 3.7 – Точність моделі(перевірка на тренувальних даних)

Бачимо, що результати вийшли дуже схожими до реальних даних, що і відображають графіки,наведені вище(Рис. 3.7 та Рис 3.8). Метрика задана таким чином, що 0 досягається лише при ідеальній роботі мережі. Бачимо, що на останніх ітераціях модель майже досягає бажаного результату, і результат варіюється від -5 до 0. Набагато важливішим в даному випадку є те, як працює модель на валідаційному наборі, бо тоді ми можемо впевнитись, що модель не лише пристосувалась до тренувального набору, а й навчилась правильно генералізувати такий тип даних.

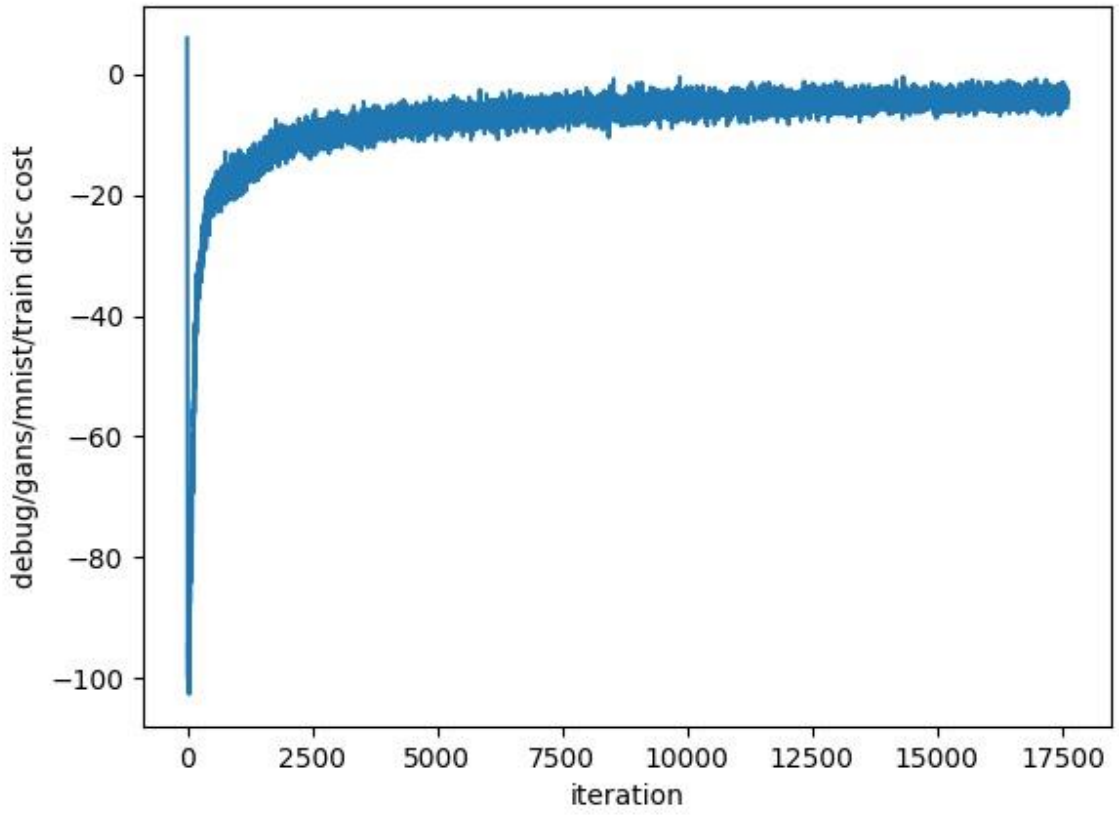

Рисунок 3.8 - Точність моделі(перевірка на валідаційних даних).

Бачимо, що на графіку валідаційного набору даних(Рис 3.8) результати майже ідентичні з тренувальним набором, отже, нейронна мережа добре генералізувала поданий набір даних, і як генератор, так і дискримінатор працюють добре,і наша модель готова для використання в якості алгоритму захисту.

3.4 Застосування натренованої мережі, як алгоритму захисту від атак «чорного ящику»

В цьому розділі ми протестуємо нашу натреновану мережу, від атак «чорного ящику» побудованих на швидкому градієнтному спуску. Як було сказано раніше, зловмисник тренує свою мережу, яка буде відрізнятись архітектурою від тої, яку він збирається атакувати. Припустимо, що він використовує обмежений набір даних, що містить в собі 150 зображень, які класифіковані моделью, на яку буде проводитись атака(тобто кожному зображенню однозначно відповідає один клас).

Таблиця 3.1 – Результати роботи ЗНМ для набору даних MNIST

Класифікатор/заміна	Не атакована	Без захисту	Захист з DefenseGAN	Захист з MagNet
А/Б	0.997	0.61	0.921	0.693
А/В	0.997	0.52	0.910	0.671
Б/А	0.963	0.49	0.88	0.59
Б/В	0.963	0.55	0.84	0.71
В/А	0.981	0.71	0.93	0.89
В/Б	0.981	0.63	0.904	0.72

Результати проведених експериментів виведено в таблицю. В експериментах ми показали точність моделі, в умовах, коли на неї зовсім не проведено атак, коли атаки проведені, але модель зовсім не захищена, коли вона захищена методом DefenseGAN, та захист за допомогою ще одного алгоритму MagNet. Можна помітити, що атака «чорного ящику», що

використовує метод швидкого градієнтного спуску досить сильно знизила точність висновків нейронної мережі. З 99 відсотків початкової точності, деякі моделі втратили майже 30 відсотків, що є надто критичною втратою, в такому простому завданні класифікації. Всі алгоритми захисту, які були запропоновані, значно покращили результати нейронної мережі, і адаптували її до вже атакованих зображень. Звісно, результати досить сильно різняться, відносно архітектур нейронних мереж. Але поскільки кожна компанія модернізує мережі так,як їм для цього зручніше, заточує кожну під своє завдання, то будемо звертати увагу на середні показники, а не на окремо обрану пару моделей. На відміну від таких методів захисту як змагальне тренування, та дистиляція нейронних мереж, DefenseGAN не залежить від архітектури мережі, і працює приблизно з однаковою точністю на всіх моделях, незалежно від параметрів та глибини.

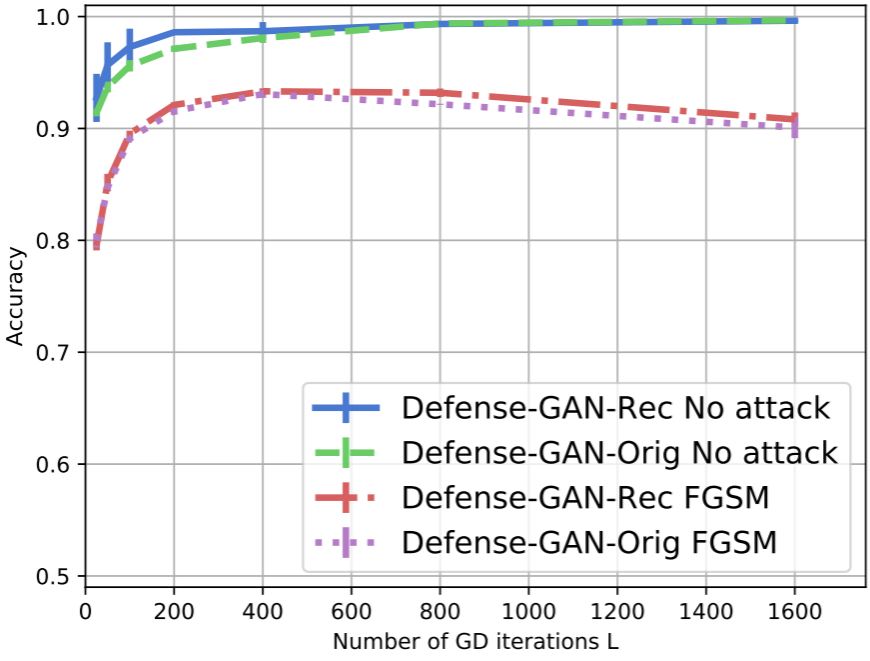

Рисунок 3.9 – Залежність результатів від кількості ітерацій швидкого градієнтного спуску

Звісно, що в залежності від того, які параметри виставити для захисту, будуть змінені результати. Таким чином отримуємо компроміс часу/розміру моделі/результатів(3.9).

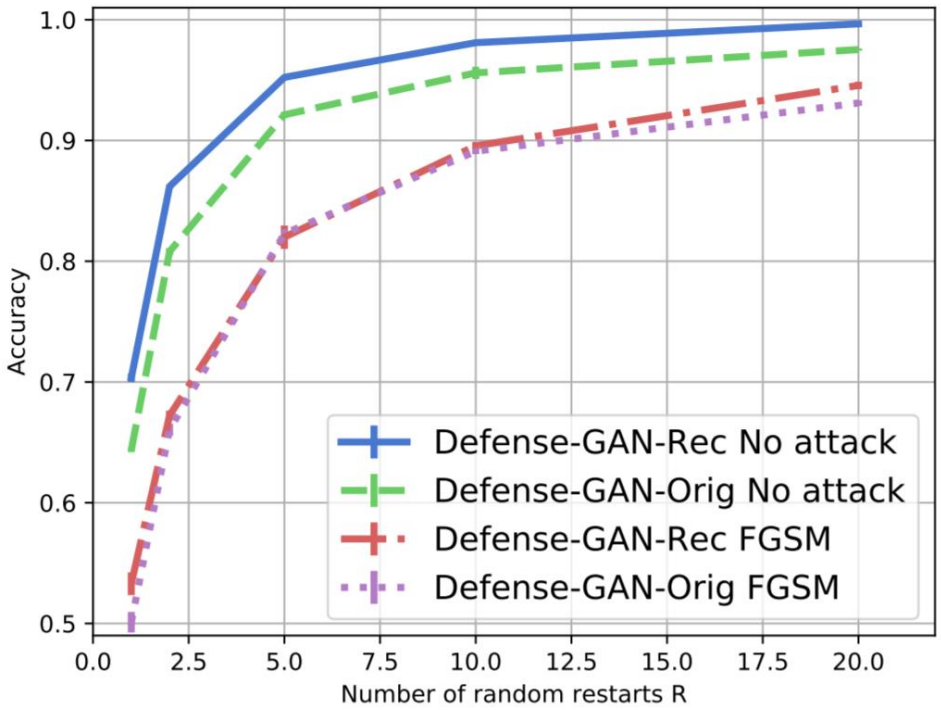

Рисунок 3.10 – Залежність результатів від кількості перезапусків тренування нейронної мережі

Інший важливий параметр – кількість перезапусків алгоритму, поскільки кожного разу він ініціалізується випадковими значеннями, і кожного разу дає різні результати(Рис.3.10).

Бачимо, що параметр R – кількість перезапусків алгоритму, є дуже важливим параметром. Здавалось, ми не змінюємо параметрів мережі, не збільшуємо її глибину, не додаємо додаткових передобробок,а просто запускаємо модель тренуватись знову. Таким нехитрим способом ми піднімаємо точність класифікації з 0.7 до 0.95, що є дуже сильним показником того, що початкова ініціалізація мережі є одним з найважливіших факторів тренування глубинних нейронних мереж.

Також були проведені експерименти з виявленням атак. Тобто ми просто використали дискримінатор нашої генеративної моделі, задля того щоб зробити бінарну класифікацію. Таким чином на виході нейронної мережі отримуємо клас 0, якщо зображення в своєму початковому вигляді, і 1, якщо зображення було змінене під впливом певних видів атак. Інтуітивно зрозуміло, що зображення до атаки має бути ближче до розподілу, що створює генератор, ніж об'єкти на які були проведені атаки. Це пов'язано з тим, що генератор був навчений створювати зображення, які нагадують початкові вхідні дані. Завдяки цьому спостереженню, ми можемо використати MSE, що є одним з видів функції втрат, як метрику, щоб зробити висновок, була проведена атака на зображення, чи воно в своєму початковому вигляді.

Таблиця 3.2 – Точність класифікації, враховуючи використання різних параметрів для зашумлення зображень

e	Результати класифікації DefenseGAN
0.10	0.987
0.15	0.982
0.20	0.964
0.30	0.93

Бачимо, що як і планувалось, модель-дискримінатор з високою точністю може передбачити чи було зображення змінене, перед тим як подаватись на вхід нейронній мережі. Тому можна запропонувати ще один вид захисту – класифікувати , було зображення атаковане чи ні, і якщо є висока ймовірність того, що зображення було змінене зловмисником, зовсім не враховувати його у подальших результатах, якщо це дозволяє специфіка задачі.

3.5 Застосування натренованої мережі, як алгоритму захисту від атак «білого ящику»

На відміну від атак «чорного ящику», в атаках «білого ящику» зловмиснику відомо набагато більше інформації, отже і атака вийде успішнішою, що знизить точність нашої мережі ще більше. Отже, сценарій атаки такий: зловмисник має повний доступ до архітектури, параметрів, градієнтів та випадкових послідовностей, що отримує на вхід генератор. Атаки, які буде використано: метод швидкого градієнтного спуску, метод додавання випадкових шумів, суміш перших двох методів та атака Карліні-Вагнера. Остання атака не згадувалась раніше, але якщо ми маємо повну інформацію про мережу, яку атакуємо, то даний тип атаки може погіршити результати класифікатору навіть до 0%. Дана атака буде проведена на сотні ітерацій, з швидкістю навчання 10, і параметром c = 100(спеціальний параметр, який використовується в атаці Карліні-Вагнера). Нажаль, ця атака потребує багато часу для підбору потрібних параметрів, для кожного окремого зображення, тому дані про результати її роботи будуть взяті з відповідної роботи.

Таблиця 3.3 - Результати роботи різних алгоритмів захисту, для атак «білого ящику»,зокрема атаки Карліні-Вагнера.

Атака	Модель	Результати	Без захисту	Defense-GAN	Mag-Net	
FGSM	A	0.997	0.217	0.988	0.243	
	B	0.962	0.022	0.956	0.142	

Продовження таблиці 3.3

	C	0.996	0.331	0.981	0.161	
	D	0.992	0.038	0.98	0.098	
RAND+FGSM	A	0.997	0.179	0.979	0.178	
	B	0.962	0.117	0.956	0.084	
	C	0.996	0.103	0.971	0.132	
	D	0.992	0.150	0.924	0.115	
Карліні-Вагнера	A	0.997	0.141	0.981	0.021	
	B	0.962	0.022	0.923	0.056	
	C	0.996	0.126	0.957	0.024	
	D	0.992	0.032	0.981	0.076	

Як бачимо з таблиці - DefenseGAN значно перевершує результати всіх інших видів захисту, хоч він і не тренувався під кожну з цих атак, а має лише генералізоване уявлення про те, як мають виглядати дані, до того як в них втручається зловмисник. Хоча ми навіть дали кожній з атак доступ до випадкового ініціювання генератору. Але слід зауважити, що результати не надто змінились, якщо у зловмисника немає такої інформації. Змагальне тренування було виконано за допомогою методу швидкого градієнтного спуску, щоб отримати атаковані приклади. Легко помітити, що змагальне тренування безсиле проти атаки Карліні-Вагнера, і не дає майже жодних покращень, в порівняння з мережею, яка зовсім не захищена.

3.6 Аналіз результатів

Після проведення всіх потрібних нам експериментів, для різного типу атак, та алгоритмів захисту, можемо підвести підсумки, та показати різницю результатів на графіках. Загалом мною було використано три типи атаки – метод швидкого градієнтного спуску, випадкове зашумлення, та атака Карліні-Вагнера. Із алгоритмів захисту використовувались – MagNet, змагальне тренування та моя модернізація DefenseGAN.

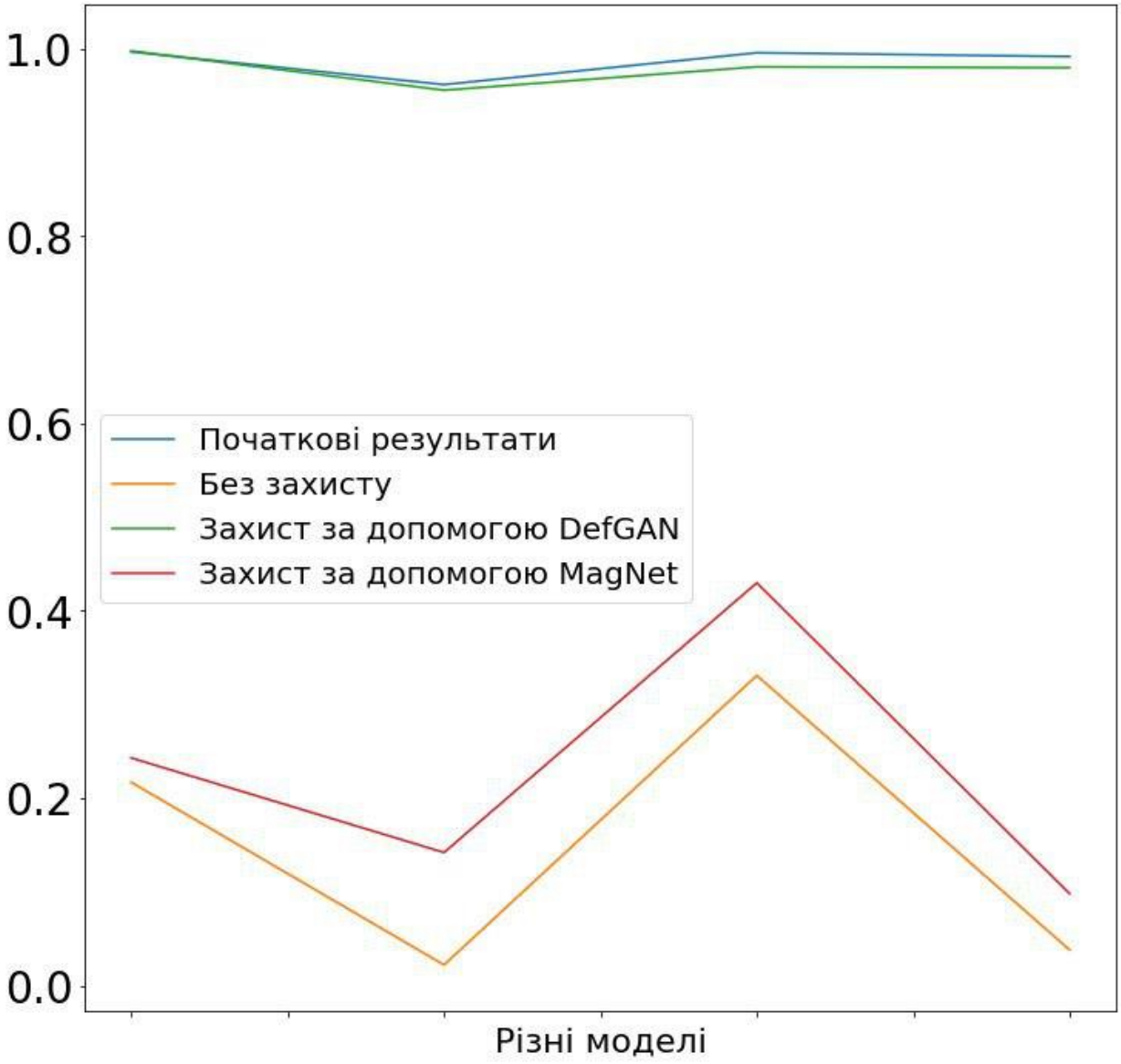

Рисунок 3.11 – Результати класифікатору в залежності від типу захисту, при атаці за допомогою швидкого градієнтного спуску

З Рис. 3.11 бачимо, що DefenseGAN показує найкращі результати, і практично повністю повторює результати нейронної мережі,до того як на неї було проведено атаку. Можемо зробити висновок, що змагальна нейронна мережа добре натренована,і навчилась знаходити і прибирати з зображення артефакти, які свідчать про використання якогось типу атаки. Таким чинном класифікатор приймає на вхід практично таке ж зображення, яким воно було спочатку, і показує дуже близьку до ідеалу точність. Слід зауважити, що результати показані для метрики топ-1(тобто точність зростає, лише якщо класифікатор з першої спроби видав правильний результат).

Отже, можна сказати, що модифікована мною архітектура DefenseGAN добре справляється з задачею захисту нейронних мереж класифікації зображень, та обходить попередні методи, що використовувались в цій області.

Висновки до розділу 3

У розділі три було розглянуто практичну реалізацію алгоритмів захисту нейронних мереж, що вирішують задачі класифікації зображень. Особливу увагу приділили тренуванню змагальної нейронної мережі на двох наборах даних – 10 класах рукописних цифр, та набору даних, що містить фото знаменитостей, і вирішує задача гендерної класифікації. Було наведено приклади роботи таких моделей, їх точність, порівняння точності на тренувальному та валідаційному датасеті, задля того щоб переконатись, що модель справді генералізує реальний розподіл вхідних даних.

Інші два підрозділи були присвячені атакам «чорного ящику» та атакам «білого ящику». Визначили з якими параметрами будемо проводити атаки, та якими алгоритмами проводити захист наших класифікаторів. Навели таблиці, з порівняльними характеристиками і побачили, що насправді, DefenseGAN справляється з задачею захисту краще всіх інших алгоритмів, особливо, коли справа стосується атак «білого ящику», а зокрема атаки Карліні-Вагнера.

Побудовані графіки та таблиці демонструють, що обраний та реалізований нами метод добре справляється з поставленою в дипломній роботі задачею, незалежно від архітектури нейронної мережі та виду атаки.

ВИСНОВКИ

Нейронні мережі дуже стрімко розвиваються в наш час, і з кожною новою атакою потребують нових методів захисту. Особливо це стосується нейронних мереж, що займаються класифікацією зображень, бо вони широко використовуються і в інших завданнях машинного навчання.

У даній роботі був проведений огляд існуючих нейронних мереж, які займаються класифікацією зображень, їх результатів. Наведені схеми їх архітектур, та пояснено, чому дана архітектура змогла покращити результати попередників.

Детальніше було розглянуто атаки на нейронні мережі, їх види, алгоритми їх побудови, і пояснено чому вони працюють на всіх відомих архітектурах нейронних мереж. Також було наведено приклади кожної атаки, з найвідоміших нині існуючих. Далі були розглянуті вже існуючі методи захисту від атак на нейронні мережі, пояснено з якими типами атак вони справляються добре,а з якими гірше, і чому потребують модернізації.

Розроблена змагальна нейронна мережа, що справляється з більшістю типів існуючих атак, та не змінює архітектуру класифікатору, а лише вносить зміни у вхідне зображення. Для реалізації поставленої задачі було використано мову Python, та бібліотеки для машинного навчанння,такі як TensorFlow, Keras та PyTorch.

Проведені експерименти показують, що розроблений метод захисту значно покращує результати вже атакованого класифікатору. Точність нейронної мережі, що класифікує зображення,в декілька разів більша, якщо використовувати її разом з розробленим методом захисту, у порівнянні з результатами мережі, яка використовується зовсім без захисту. Також покращені результати вже існуючих алгоритмів захисту,таких як MagNet та змагальне тренування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Knagg O. Know your enemy [Електронний ресурс] / Oscar Knagg. – 2019. – Режим доступу до ресурсу: https://towardsdatascience.com/know-your-enemy-the-fascinating-implications-of-adversarial-examples-5936bccb24af.
2. Future of Driving [Електронний ресурс] – Режим доступу до ресурсу: https://www.tesla.com/autopilot.
3. Joseph A. Application of Neural Network in User Authentication for Smart Home System [Електронний ресурс] / A. Joseph. – 2009. – Режим доступу до ресурсу: https://waset.org/publications/9242/application-of-neural-network-in-user-authentication-for-smart-home-system.
4. Seif G. Deep Learning for Image Recognition: why it's challenging, where we've been, and what's next [Електронний ресурс] / George Seif. – 2018. – Режим доступу до ресурсу: https://towardsdatascience.com/deep-learning-for-image-classification-why-its-challenging-where-we-ve-been-and-what-s-next-93b56948fcef.
5. Image Classification [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://shodhganga.inflibnet.ac.in/bitstream/10603/24460/10/10_chapter5.pdf.
6. Krizhevskiy A. ImageNet Classification with Deep Convolutional Neural Networks [Електронний ресурс] / A. Krizhevskiy, I. Sutskever, G. Hinton. – 2016. – Режим доступу до ресурсу: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.
7. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://arxiv.org/pdf/1409.1556.pdf.
8. He K. Deep Residual Learning for Image Recognition [Електронний ресурс] / Kaiming He. – 2015. – Режим доступу до ресурсу: https://arxiv.org/pdf/1512.03385.pdf.

9. Huang G. Densely Connected Convolutional Networks [Електронний ресурс] / G. Huang, Z. Liu. – 2018. – Режим доступу до ресурсу: https://arxiv.org/pdf/1608.06993.pdf.
10. Chatel G. Adversarial examples in deep learning [Електронний ресурс] / Gregory Chatel. – 2017. – Режим доступу до ресурсу: https://towardsdatascience.com/adversarial-examples-in-deep-learning-be0b08a94953.
11. Tsui K. Perhaps the Simplest Introduction of Adversarial Examples Ever [Електронний ресурс] / Ken Tsui. – 2018. – Режим доступу до ресурсу: https://towardsdatascience.com/perhaps-the-simplest-introduction-of-adversarial-examples-ever-c0839a759b8d.
12. Su J. One Pixel Attack for Fooling Deep Neural Networks [Електронний ресурс] / Jiawei Su. – 2019. – Режим доступу до ресурсу: https://arxiv.org/pdf/1710.08864.pdf.
13. Xiao C. Generating Adversarial Examples with Adversarial Networks [Електронний ресурс] / C. Xiao, J. Zhu. – 2019. – Режим доступу до ресурсу: https://arxiv.org/pdf/1801.02610.pdf.
14. Carlini N. Towards Evaluating the Robustness of Neural Networks [Електронний ресурс] / N. Carlini, D. Wagner. – 2017. – Режим доступу до ресурсу: https://nicholas.carlini.com/papers/2017_sp_nnrobustattacks.pdf.
15. Goodfellow J. Generative Adversarial Nets [Електронний ресурс] / Jan Goodfellow. – 2014. – Режим доступу до ресурсу: https://arxiv.org/pdf/1406.2661.pdf.
16. A Beginner's Guide to Generative Adversarial Networks (GANs) [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://skymind.ai/wiki/generative-adversarial-network-gan.

17. Samangouei P. DEFENSE-GAN: PROTECTING CLASSIFIERS AGAINST ADVERSARIAL ATTACKS USING GENERATIVE MODELS [Електронний ресурс] / Pouya Samangouei. – 2018. – Режим доступу до ресурсу: https://openreview.net/pdf?id=BkJ3ibb0-.