Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Курсова робота
освітній ступінь – магістр

на тему: **«ЗМАГАЛЬНІ ПРИКЛАДИ ТА ЇХ ЗНАХОДЖЕННЯ В ЗАДАЧАХ ОБРОБКИ ЗОБРАЖЕНЬ»**

Виконала: студентка 1-го року навчання
освітньої програми «Прикладна математика»,
спеціальності 113 Прикладна математика

Фісун Єлизавета Борисівна

Керівник: Крюкова Г.В.
доцент

Рецензент: .

Курсова робота захищена
з оцінкою _____

Секретар ЕК_____
(підпис)

«_____» _____ 20___р.

Київ - 2023

Зміст

Вступ

В останні роки швидкий розвиток алгоритмів машинного навчання та глибоких нейронних мереж здійснив революцію в галузі комп'ютерного зору та обробки зображень. Ці алгоритми досягли чудової продуктивності в різних завданнях, починаючи від класифікації зображень і закінчуючи виявленням об'єктів. Однак разом із цими досягненнями виникла нова проблема: вразливість моделей глибокого навчання до змагальних прикладів. Змагальні приклади — це ретельно розроблені вхідні дані, які вводять модель в оману, хоча людському оку складно їх відрізнити від початкових даних. Це викликає критичні питання щодо стійкості, надійності та безпеки моделей машинного навчання, особливо в критично важливих для безпеки програмах, таких як автономні транспортні засоби, медична діагностика та кібербезпека. Змагальні атаки створюють потенційні ризики в різних областях, включаючи, але не обмежуючись системами розпізнавання зображень, де наслідки неправильної класифікації можуть бути згубними. Оскільки розгортання систем машинного навчання стає все більш поширеним, важливо усунути вразливі місця цих моделей, щоб забезпечити їх практичне та надійне використання в реальних сценаріях. За мету даної роботи були поставлені такі завдання: розібратися з що таке змагальні приклади та атаки, визначити які вони бувають та методи їх пошуку, а також застосувати ці змагальні зображення та атаки до конкретних задач (класифікації, сенментації тощо.)

1 Основні поняття та алгоритми

1.1 Задачі комп'ютерного зору

Класифікація зображень — це процес присвоєння міток зображенням відповідно до їх типів (класів). Існують два типи класифікації зображення: класифікація з однією міткою та класифікація з кількома мітками. Перша є найпоширенішим завданням класифікації зображень у випадку навчання зі вчителем (supervised learning), як випливає з назви, одна мітка або анотація присутня для кожного зображення в класифікації. Таким чином, модель виводить одне значення або прогноз для кожного зображення, яке вона бачить. Вихідні дані моделі є вектором, довжина якого дорівнює числу класів, і значенням, що вказує на приналежність зображення до цього класу. Класифікація з однією міткою може мати тип бінарної чи багатокласової класифікації. Класифікація за кількома мітками — це завдання класифікації, де кожне зображення може містити більше однієї мітки, а деякі зображення можуть містити всі мітки одночасно. Завдання класифікації з декількома мітками широко існують у галузі медичної візуалізації, де у пацієнта може бути більше одного захворювання, яке можна діагностувати на основі візуальних даних у формі рентгенівських знімків. [1]

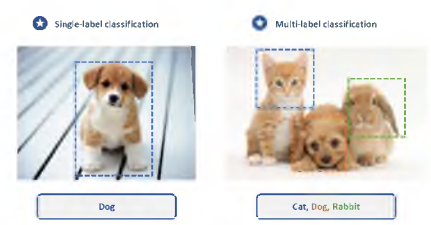


Рис. 1: Single vs multiple image classification

Сегментація зображення - це метод поділу цифрового зображення на підгрупи, які називаються сегментами зображення, що зменшує складність зображення та забезпечує подальшу обробку чи аналіз кожного сегмента зображення. Технічно сегментація зображення – це функція, яка приймає вхідне зображення та повертає маску або матрицю з різними елементами, що визначають клас об'єкта, до якого належить кожен піксель. Сегментація зображення зазвичай використовується для виявлення об'єктів. Замість обробки всього зображення поширеною практикою є спочатку використання алгоритму сегментації зображення для пошуку цільових об'єктів зображення. Тоді детектор об'єктів може працювати з обмежувальною рамкою, уже визначеною алгоритмом сегментації. Завдяки цьому йому не потрібно обробляти все зображення, що покращує точність і скорочує

час прознозу. Цей метод має широке звстосування, включаючи аналіз медичних зображень, комп'ютерний зір для автономних транспортних засобів, розпізнавання та виявлення облич, відеоспостереження та аналіз супутникових зображень. Існує декілька видів сегментації, основними з яких є: семантична сегментація, сегментація екземпляра та паноптична сегментація. Для того щоб зрозуміти різницю між цими підходами введемо наступні означення: матеріалом *(stuff)* називатимемо те, що не можна порахувати, наприклад, небо і дорога, а речима *(things)* - фактичні об'єкти на зображенні, які можна порахувати, призначивши різні ідентифікатори екземплярів кожному з них. Семантична сегментація включає впорядкування пікселів у зображенні на основі семантичних класів. У цій моделі кожен піксель належить до певного класу, а модель сегментації не посилається на будь-який інший контекст або інформацію. Наприклад, семантична сегментація, виконана на зображенні з кількома деревами та транспортними засобами, забезпечить маску, яка класифікує всі типи дерев в один клас (дерево), а всі типи транспортних засобів, будь то автобуси, автомобілі чи велосипеди, в один клас(транспортні засоби). Сегментація екземплярів класифікує пікселі за категоріями на основі об'єктів (things), а не класів. Алгоритм сегментації екземпляра не має уявлення про клас, до якого належить класифікована область, але може відокремлювати перекриваючі або дуже подібні області об'єктів на основі їхніх меж. Наприклад, припустімо, що модель обробляє зображення багатолюдної вулиці, тоді у випадку сегментації екземпляру помічаються окремі об'єкти в натовпі, а також визначається кількість екземплярів на зображенні. Паноптична сегментація (things+stuff) — це новий тип сегментації, який часто виражається як поєднання семантичної та екземплярної сегментації. Він передбачає ідентичність кожного об'єкта, відокремлюючи кожен екземпляр кожного об'єкта на зображенні. Наприклад, безпілотні автомобілі повинні швидко й точно знімати та розуміти навколишнє середовище. Вони можуть досягти цього, подаючи живий потік зображень на алгоритм панорамної сегментації. [1]

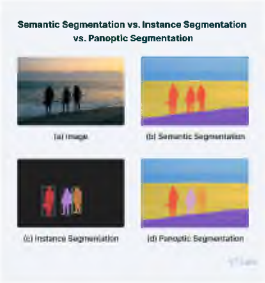


Рис. 2: Different types of segmentation

1.2 Adversarial example

Adversarial example — це модифіковані вхідні дані, які створені з метою заплутати нейронну мережу, що призводить до неправильної класифікації з досить високою впевненістю. Ці спотворені вхідні дані неможливо розрізнити людським оком, але призводять до того, що мережа не може ідентифікувати справжній вміст зображення. Формально, adversarial example може бути визначений наступним чином: датасет $\{x_i, y_i\}_{i=1}^{N}$, де x_i - вхідне зображення, y_i - відповідний клас об'єкта, N - розмір датасету. Нейронна мережа $f(\cdot)$ яка приймає на вхід x та повертає передбаченя $f(x)$. Відповідна функція втрат (adversarial loss) позначається $J(\theta, x, y)$, де θ ваги моделі. Перехресна ентропія між $f(x)$ та класом (в числовому представленні) визначається як функція втрат $J(f(x), y)$. Вхідні дані x' вважатимемо спотвореним зображенням,

$$x' : D(x, x') < \eta, f(x') \neq y \tag{1}$$

де $D(\cdot, \cdot)$ метрика відстані, η - константа, що обмежує відстань та показує дозволений рівень модифікації зображення. [2]

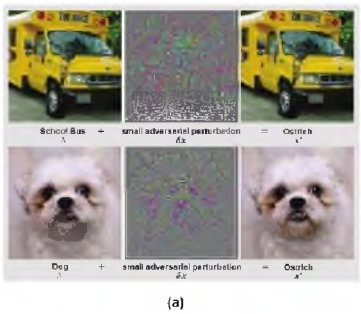


Рис. 3: Adversarial example

1.3 Вимірювання збурень

За означенням збурене зображення x' має бути досить близьким до початкового ("чистого") x, непомітним для людського зору, найчастіше використовуються $p - norms$ для контролю розміру та кількості збурень, які додаються до вхідного зображення. L_p обчислює відстань $\|x - x'\|_p$ між вхідним зображенням x та модифікованим x', де $p \in \{0, 1, 2 \ldots \infty\}$.

$$L_p = \sqrt[p]{\sum \|x - x'\|^p} \tag{2}$$

Якщо $p = 0$, то норма виражає кількість змінених пікселів для генерації adversarial example. L_2 це стандартна Евклідова відстань. У випадку, якщо L_∞ вимірює

максимум різниці серед усіх пікселів між двома зображеннями, $L_p = \|x-x'\|_\infty = max(|x_1 - x'_1|, |x_2 - x'_2|, \ldots, |x_n - x'_n|)$. [3]

1.4 Класифікація атак

У контексті безпеки adversarial attacks та зловмисники класифікуються за моделями загроз [3]. Модель загрози можна поділити на три основні компоненти: знання зловмисника про модель (параметри та архітектура), мету атаки та яким чином атака буде виконуватись. Тоді модель загрози можна класифікувати за п'ятьма різними напрямками: (i) вплив зловмисника, (ii) знання зловмисника, (iii) порушення безпеки, (iv) специфічність атаки, (v) підхід до атаки.

Вплив зловмисника Згідно з Хіао [4] нападник може проводити два типи атак беручи до уваги вплив на класифікаційну модель: причинні(або отруюючі) атаки та атаки ухилення (або пошукові).

Причинні(або отруюючі) атаки в цьому сценарії нападник має вплив на модель під час стадії її тренування, а саме тренувальна вибірка "забруднюється"adversarial examples з метою зробити класифікаційну модель несумісною з справжнім розподілом даних.

Атаки ухилення (або пошукові) на відміну від причинних атак, зловмисник впливає на моделі глибокого навчання на стадії тестування. Атаки ухилення є найпоширенішим типом атак, коли зловмисник створює adversarial examples, які призводять до неправильної класифікації моделей глибокого навчання, як правило, з високою впевненістю прогнозу. Атаки ухилення також можуть мати дослідницький характер, коли метою зловмисника є збір інформації про цільову модель, таку як її параметри, архітектури, функції втрат тощо. Найпоширеніша дослідницька атака — це атака введення/виведення, коли зловмисник надає цільовій моделі створені ним adversarial examples. Після цього, він спостерігає на результат цільової моделі та намагається відтворити сурогатну (замісну) модель таку щоб вона була схожа до цільової. Атака введення/виведення зазвичай є першим кроком для проведення black-box атак.

Знання зловмисника - рівень доступу інформації про модель, яку має нападник. Існують два типи атак за цим критерієм: white-box attack and black-box.

White-box attacks - в цьому сценарії нападник має повний доступ до моделі, включаючи параметри захисту та архітектуру. Такі атаки трапляються найменш часто в реальному світі через застосування заходів захисту (таких як контроль користувачів наприклад), для того щоб запобігти несанкціонований доступ до системи. На противагу, white-box attacks вважаються найбільш потужними та з цієї причини зазвичай використовуються для оцінки надійності засобів захисту, але, розробка засобів захисту, які будуть стійкими до такого виду атак поки що є відкритою проблемою.

Black-box attacks в цьому випадку, нападник не має доступу та знань, ні до класифікаціїної моделі, ні до методу захисту. Black-box attacks накладають багато обмежень для зловмисників, незважаючи на складність відтворення атаки, нападник все одно може заплутати цільову модель завдяки можливості передачі adversarial examples.

Порушення безпеки асоціюється з ціллю нападника при проведенні атаки проти класифікатора. Розрізняють три типи порушень, які можуть вплинути на (i) цілісність, (ii) доступність та (iii) конфіденційність цільових класифікаторів.

Порушення цілісності відбувається коли adversarial examples, створені зловмисником, здатні непомітно обійти існуючі контрзаходи та привести цільову модель до неправильної класифікації, але без шкоди для функціональності системи

Порушення доступності виникає, коли функціональність системи також порушена, що спричиняє відмову в обслуговуванні. Порушення доступності головним чином впливає на надійність системи навчання, підвищуючи невизначеність прогнозів.

Порушення конфіденційності відбувається, коли зловмисник може отримати доступ до відповідної інформації щодо цільової моделі, такої як її параметри, архітектура та використані алгоритми навчання. Порушення конфіденційності в глибокому навчанні сильно пов'язані з black-box attacks, коли зловмисник запитує цільову модель, щоб перепроектувати її та створити сурогатну модель, яка створює adversarial examples ближче до вихідного розподілу даних.

Специфічність атаки бувать цілеспрямовані та нецілеспрямовані атаки. Для цілеспрямованої атаки, adversarial image генерується щоб змусити модель неправильно класифікувати їх в попередньо визначеному класі. Наприклад: нехай x - вихідне правильне зображення, y - правильний клас для зображення x та f-класифікатор; тоді adversarial example $x' = x + \delta x$. Для цільової атаки нападник шукає таке збурення δx щоб модель f повернула специфічний клас y', так що $f(x + \delta x) = y'$, $y' \neq y$. І навпаки, під час нецільової атаки генерується adversarial image x', такий що $f(x) \neq f(x')$. Цільові атаки зазвичай спричиняють вищі обчислювальні витрати порівняно з нецільовими атаками.

Підхід до атаки - алгоритм, який був використаний для створення збурення. Можна виділити три основні підходи, що грунтуються на: (i) градієнті, (ii) можливостіь передачі/оцінці, (iii) рішенні.

В алгоритмах основаних на градієнті використовується детальна інформація цільової моделі щодо її градієнта. Цей підхід до атаки зазвичай використовується у white-box attacks, коли зловмисник має повні знання та доступ до моделі.

Атаки на основі трансферу/оцінки: ці алгоритми атаки залежать або від отримання доступу до даних, що використовується цільовою моделлю, або оцінки, передбачені нею для наближення градієнта. Зазвичай результати, отримані шля-

хом запиту до цільової нейронної мережі, використовуються як оцінки. Потім ці бали використовуються разом із навчальним набором даних, щоб відповідати сурогатній моделі, яка буде створювати збурення, які будуть вставлені в вихідні зображення. Цей підхід до атаки є часто корисний під час black-box attack.
Атаки на основі прийняття рішень цей підхід вперше запровадили Brendel та ін. автори [5] вважають більш простим і гнучким підходом, оскільки потребує незначних змін параметрів, ніж атаки на основі градієнта. Атака на основі прийняття рішень зазвичай робить запит до softmax рівня цільової моделі та, ітеративно, обчислює менші збурення за допомогою процесу відбору семплів(rejection sampling).

1.5 Алгоритми для генерації adversarial examples

У комп'ютерному зорі алгоритми, що використовуються для генерування adversarial examples, є методами оптимізації, які зазвичай досліджують недоліки узагальнення в попередньо навчених моделях, щоб створити та додати збурення в вихідні зображення. Розглянемо однопіксельну атаку, яка відноситься до black-box attack [6]. Для генерації однопіксельних збурень використовується алгоритм диференціальної еволюції. У випадку black-box attack знання нападника про модель дуже обмежені, а саме нам відомо лише ймовірності класів (probability labels), але невідома внутрішня структура моделі (градієнти). Тому даний тип атаки є досить гнучким, адже може бути застосований для багатьох мереж глибокого навчання (навіть для тих які не є диференційованими або процес обчислення градієнту є складним). Генерація adversarial images може бути сформульована як оптимізаційна задача з обмеженнями. Нехай f цільовий класифікатор, який приймає n розмірні вектори, де $x = (x_1, \ldots, x_n)$ є початковим зображенням якому відповідає правильно класифікований клас t. Ймовірність, що x належить до класу t позначатимемо як $f_t(x)$. Вектор $e(x) = (e_1, \ldots, e_n)$ це є збурення до початкового зображення x. Метою нападника у випадку цільової атаки є знайти оптимальне рішення $e(x)^*$ для наступної задачі:

$$\begin{gathered} \max_{e(x)^*} \ f_{adv}(x + e(x)) \\ \text{за умови що } ||e(x)||_0 \leq d \end{gathered} \tag{3}$$

де d це маленьке число, в нащому випадку це 1. Однопіксельну модифікацію можна розглядати як збурення точки даних уздовж напрямку, паралельного осі одного з n вимірів. На даній ілюстрації зображена одна та двох піксельна атака в трьох вимірному вхідному просторі (зображення має три пікселі). Зелена

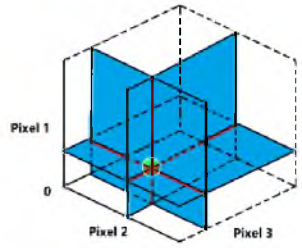


Рис. 4: One and two pixel attacks on 3-dim input space

точка(сфера) виражає початкове зображення. У випадку однопіксельного збурення простір пошуку це три перпендикулярні лінії які перетинаються в зеленій точці, які позначені червоно-чорними смугами. Якщо розглянути двохпіксельну атаку, простір пошуку це три сині двовимірні площини.
Диференціальна еволюція(Storn & Price, 1997) [7]- метод багатовимірної математичної оптимізації, що відноситься до класу стохастичних алгоритмів оптимізації (тобто працює з використанням випадкових чисел) і використовує деякі ідеї генетичних алгоритмів. У його базовому вигляді алгоритм можна описати таким чином.

1) Спочатку генерується деяка множина векторів, так зване покоління. Під векторами розуміються точки n-вимірного простору, в якому визначена цільова функція $f(x)$, яку потрібно максимізувати(мінімізувати).
2) На кожній ітерації алгоритм генерує нове покоління векторів (children), випадковим чином комбінуючи вектори з попереднього покоління. Число векторів в кожному поколінні одне й те саме і є одним з параметрів методу. Нове покоління векторів генерується в такий спосіб. Для кожного вектора x_i зі старого покоління (parents) вибираються три різних випадкових вектори x_{r1}, x_{r2}, x_{r3} та генерується мутантний вектор v_i за формулою:

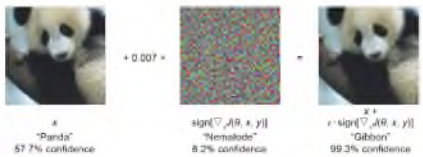
$$v_i = x_{r1} + F(x_{r2} - x_{r3}) \tag{4}$$

де F це дійсна позитивна константа в інтервалі [0,2], що виражає параметр мутаціїї, коефіцієнт масштабу.
3) Над мутантним вектором виконується операція "схрещування"(crossover), яка

полягає в тому, що деякі його координати заміщаються відповідними координатами з початкового вектора x_i (кожна координата заміщається з деякою ймовірністю, яка також є ще одним з параметрів цього методу).
4) Отриманий після схрещування вектор називається пробним вектором (trial vector). Якщо він виявляється кращим за вектор x_i, то в новому поколінні вектор x_i замінюється на пробний вектор, а в іншому разі — залишається x_i.
В статті [6] збурення представляються як вектори з 5-ма координатами (x, y R, G, B). Випадковим чином ініціалізується 400 збурень та обчислюється цільова функція (ймовірність класу для зміненого зображення). Параметр мутації $F = 0.5$. З загальної схеми алгоритму диференціальної еволюції пропускається етап схрещування. Згенеровнаий новий кандидат змагається з своїм відповідником зі старого покоління, переможешь переходить на наступну ітерацію. максимальна кількість ітерацій - 100 та критерій ранньої зупинки буде застосований у випадку якщо ймовірність цільового класу перевищує 90% (цільова атака), а також якщо ймовірність нижча за 5% у разі нецільової атаки. В кінці ймовірність правильного класу порівнюється з найбільшою ймовірністю помилковим класу для того щоб визначити успішність атаки.
Метод швидкого градієнтного знака (FGSM) Godfellow et al. [8] вперше запропонував ефективний метод нецільової атаки FGSM, для генерації adversarial examples з L_∞ метрикою. FGSM це типовий однокроковий метод атаки, він обчислює градієент функції втрат $J(\theta, x, y)$, а потім використовує знак цього градієнт адля створення adversarial example. Градієнти беруться відносно вхідних зображень, оскільки метою є створення зображення, яке максимізує функцію втрат. Це робиться шляхом визначення того, який внесок кожного пікселя у значення функції втрат, а методи додають збурення відповідно. Формально можна визначити таким чином:

$$x' = x + \epsilon \cdot sign[\nabla_x J(\theta, x, y)] \tag{5}$$

де ϵ величина збурення.



1.6 Adversarial attack on semantic segmentation model

Загалом adversarial examples можуть бути також застосовані для задач семантичної сегментації та детекції об'єктів, що є значно складнішими ніж задача

класифікації. Для виконання атаки на цих задачах потрібно оптимізувати функцію втрат на множині цілей (targets). Розглянемо Dense Adversary Generation (DAG) алгоритм, який враховує всі цілі одночасно та оптимізує загальну функцію втрат [9]. Нехай X це зображення, яке містить N цілей розпізнавання $\mathcal{T} = t_1, t_2, \ldots, t_N$. Кожній цілі $t_n, n = 1, 2, \ldots, N$ присвоєна мітка правильного класу $l_n \in 1, 2, ..., C$, де C - це кількість класів в датасеті, позначатимемо цю множину $\mathcal{L} = l_1, l_2, \ldots, l_n$. Для задачі сегментації множина τ складається з усіх пікселів зображення. Нехай $f(X, t_n) \in \mathcal{R}^C$ визначає вектор оцінки класифікації (до нормалізації softmax) на n-й цілі розпізнавання X. Для того щоб згенерувати adversarial example потрібно щоб передбачення для всіх цілей були неправильними, тобто $\forall n,\ argmax_c f_c(X + r, t_n) \neq l_n.r$ означає збурення, що додається до зображення X. Також нам потрібно визначити хибні мітки l'_n для кожної цілі, де l'_n обирається випадковим чином з множини інших класів, тобто $l'_n \in 1, 2, \ldots, C\{l_n$, позначатимемо цю множину як $\mathcal{L}' = l'_1, l'_2, \ldots, l'_n$. Практично це функція перестановки π: $1, 2, \ldots, C \Rightarrow 1, 2, \ldots, C$ для кожного зображення незалежно, де $\pi(c) \neq c$ для $c = 1, 2, \ldots, C$. Визначимо функцію втрат наступним чином:

$$L(X, \mathcal{T}, \mathcal{L}, \mathcal{L}') = \sum_{n=1}^{N} [f_{l_n}(X, t_n) - f_{l'_n}(X, t_n)] \tag{6}$$

Мінімізація L може бути досягнута таким чином якщо кожна ціль буде неправильно передбачена, зменшення впевненості в вхідному правильному класі $f_{l_n}(X, t_n)$ та одночасне збільшення бажаного (неправильного) класу $f_{l'_n}(X, t_n)$. Для оптимізації застосовується алгоритм градієнтного спуску.

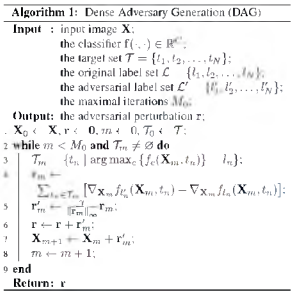


Рис. 5: Dense Adversary Generation

Алгоритм зупиняє свою роботу якщо всі цілі передбачені як заплановано або він досягає максимальної кількості ітерацій, яка є 200 для задачі сегментації.

1.7 Transferability of adversarial examples

Adversarial sample transferability це властивість така що, adversarial examples були спеціально створені для введення в оману спеціальну модель f можуть вводити в оману інші моделі f' навіть якщо їх архітектури значно відрізняються. Таким чином, зловмисник може навчити свою власну сурогатну модель створювати adversarial examples проти неї та передавати їх "моделі жертв", маючи дуже мало інформації про неї. Формально adversarial sample transferability можна визначити таким чином [10]:

$$\Omega_X(f, f') = |f'(x) \neq f'(x + \delta) : x \in X| \tag{7}$$

де множина X відображає очікуваний розподіл вхідних даних, що розв'язується моделями. Дану властивість можна розділити на два типи: Intra-technique transferability та Cross-technique transferability. У першому випадку, моделі f та f' тренуються використовуючий один і той самий алгоритм машинного навчання, але різні параметри ініціалізації чи датасет. Другий тип передбачає що моделі f та f' використовують різні техніки машинного навчання, наприклад нейронна модель та SVM алгоритм. У статті [10] розглядаються та підтверджуються дві гіпотези: "Обидва типи adversarial sample transferability є незмінним феноменом у просторі методів машинного навчання"та "Black-box attacks можливі в практичних умовах проти будь-якого невідомого класифікатора машинного навчання". Розглянемо результати, що були отримані при тестуванні гіпотез.

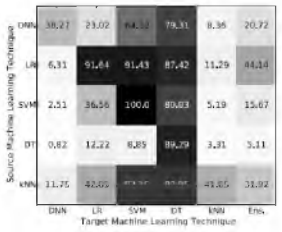


Рис. 6: Cross-technique transferability

Значення в кожній комірці відображає відсоток успішно створених adversarial samples з (MNIST dataset), які також атакували цільову модель та змінили мітку класу. Зліва зображена сурогатна модель (модель заміни) та цільова модель знизу. Загалом чим ближча архітектура сурогатної моделі до архітектури цільової моделі, тим ймовірніша атака. Моделі логістичної регресії (LR) є хорошою моделлю заміни для інших логістичних регресій, методу опорних векторів (SVM) та дерев рішень (Decision Trees). Дерева рішень (Decision trees) є найбільш вразливими до атак, оскільки атаки з будь-якої архітектури добре переносяться. Набільш стійкою архітектурою до атак виявилась Глибока нейронна мережа (Deep neural network).

2 Приклади застосування атак

2.1 Image classification task

Для задачі класифікації зображення було обрано датасет CIFAR-10 (Canadian Institute For Advanced Research) [11], який складається з 60,000 тисяч кольорових картинок, розміром 32x32 в 10 класах, по 6000 картинок в кожному класі. Тренувальна вибірка містить 50,000 картинок, а тестова - 10,000. 10 різних класів предствляють літаки, автомобілі, птахів, котів, оленів, собак, жаб, коней, кораблі та вантажівки. Розгланемо декілька картинок з кожного класу. Важливим етапом перед побудовою та тренуванням моделі є підготовка даних,

Рис. 7: CIFAR-10 (перші 5 класів)

в даному випадку я застосувала нормалізацію даних та конвертацію в унітарний код.
1) Нормалізація даних це етап попередньої обробки в машинному навчанні, який робить навчання менш чутливим до масштабу функцій. Це дозволяє нашій моделі вивчати й оновлювати свої ваги ефективніше, що призводить до швидшого часу навчання та підвищення продуктивності. У нашому випадку ми ділимо зображення на 255, таким чином отримуємо пікселі в діапазоні (0,1).
2) Конвертація класових векторів в бінарні матриці (унітарне кодування one-hot encoding). Наприклад клас 3, що відповідає коту, ми перетворюємо у вектор [0,0,0,1,0,0,0,0,0,0]. Причина, по якій ми це робимо, полягає в тому, що це дозволяє нам використовувати певні типи функцій втрат, як-от категорійну перехресну ентропію, яка очікує, що мітки будуть надані в цьому форматі. Функція втрат категорійної перехресної ентропії — це функція втрат, яка використовується в задачах багатокласової класифікації. Ця функція втрат базується на порівнянні між справжньою ймовірністю і прогнозованою ймовірністю для кожного класу на виході моделі.
Для виконанання задачі класифікації було обрано CNN модель, яка дала такі результати loss: 0.66, accuracy: 0.84. Розглянемо матрицю невідповідностей (confusion matrix). Кожен рядок матриці відповідає фактичному класу, а кожен стовпець відповідає прогнозованому класу. Діагональні комірки (від верхнього лівого до нижнього правого) представляють кількість точок, для яких прогнозована мітка дорівнює справжній мітці (тобто правильно класифіковані точки

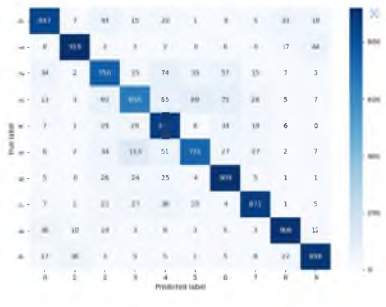


Рис. 8: Confusion matrix

для кожного класу). Недіагональні комірки забезпечують неправильну класифікацію - іншими словами, комірка в i-му рядку та j-му стовпці представляє кількість спостережень, які, як відомо, належать до групи i, але передбачувано належать до групи j. Темніші кольори зазвичай представляють вищі числа, тому найтемніші квадрати вздовж діагоналі є хорошим знаком, який вказує на багато правильних прогнозів, а темні квадрати поза діагоналлю навпаки показують загальні неправильні класифікації.

Після того як ми натренували модель, потрібно згенерувати adversarial example та зробити однопіксельну атаку, застосовуючи алгоритм диференціальної еволюції. Для цього визначимо три функції: model loss, яка повертає впевненість для присвоєння неправильної мітки для adversarial example, perturb image, яка повертає зображення зі зміненим значенням пікселя, так зване збурене зображення (adversarial example) та one pixel attack яка використовує функцію differential evolution. На ілюстрації зображено оригінальне зображення та збурене (червоним квадратиком обведений модифікований піксель).

Після проведеної атаки модель дала такий результат з дуже високою ймовірністю вона класифікувала клас кінь(7) як автомобіль(1).

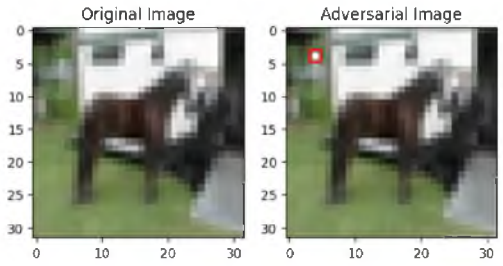


Рис. 9: One-pixel attack

Висновки

Спочатку для застосування adversarial attack на прикладах я обрала MNIST dataset - картинки рукописних цифр. Була побудована та натренована CNN, її точність була дуже високою, майже 0.98. Але однопіксельна атака не була успішною. Однією з причин такого результату може бути те, що картинки невеликі 28x28 та мають всього один канал (чорно-білі), тобто це означає що зміна всього лише одного пікселя може істотно не змінити прогноз моделі. Окрім цього, модель може бути просто стікою до такого виду атаки, адже вона добре навчена і може добре обробляти незначні варіації вхідних даних. CNN все одно може розпізнати цифру, навіть якщо один піксель змінено, особливо якщо піксель знаходиться в менш важливому місці (наприклад, біля краю зображення або в частині зображення, яка не є критичною для розпізнавання цифри). Але потім я спробувала повторити атаку на іншому датасеті (cifar10) та іншій моделі і вона виявилась успішною. Результат прогнозу моделі змінив мітку зображення з досить високою впевненістю. Загалом на однопіксельні атаки впливає багато факторів: сам датасет, складність архітектури моделі, параметри алгоритму диференціальної еволюції, а також варто пам'ятати що це black-box атака, тобто нападник знає дуже мало інформації про модель.

Література

[1] Richard Szeliski. Computer Vision: Algorithms and Applications 2nd Edition. 2010

[2] Kui Ren, Tianhang Zheng, Zhan Qin, Xue Liu. Adversarial Attacks and Defenses in Deep Learning. 2020

[3] Gabriel R. Machado, Eugento Silva, Ronaldo R. Goldschmidt. Adversarial Machine Learning in Image Classification: A Survey Towards the Defender's Perspective. 2020

[4] Huang Xiao. 2017. Adversarial and Secure Machine Learning. Ph.D. Dissertation. Universit?t M?nchen. https://mediatum.ub.tum.de/1335448 04 de fevereiro de 2019.

[5] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. arXiv preprint arXiv:1712.04248 (2017).

[6] Jiawei Su, Danilo Vasconcellos Vargas and Kouichi Sakurai. One Pixel Attack for Fooling Deep Neural Networks. 2019

[7] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4): pp.341–359, 1997.

[8] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In International Conference on Learning Representations. http://arxiv.org/abs/1412.6572

[9] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie1, Alan Yuille. Adversarial Examples for Semantic Segmentation and Object Detection.

[10] Nicolas Papernot and Patrick McDaniel, Ian Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. 2016

[11] https://www.cs.toronto.edu/ kriz/cifar.html