МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кваліфікаційна наукова праця на правах рукопису

ІВАНЮК АНДРІЙ ОЛЕГОВИЧ

УДК 519.688:004.934

ДИСЕРТАЦІЯ

ДОСЛІДЖЕННЯ ВЗАЄМОЗВ'ЯЗКІВ У ДАНИХ З ВИКОРИСТАННЯМ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

113 «Прикладна математика»

11 «Математика та статистика»

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ А. О. ІВАНЮК

Науковий керівник: Крюкова Галина Віталіївна,

кандидат фізико-математичних наук, доцент

Електронний підпис
ІВАНЮК
АНДРІЙ
ОЛЕГОВИЧ
Ідентифікаційний
код
3554413113
Підписано у Вчасно

Київ — 2024

АНОТАЦІЯ

Іванюк А.О. Дослідження взаємозв'язків у даних з використанням штучних нейронних мереж — Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії у галузі знань 11 "Математика та статистика" за спеціальністю 113 "Прикладна математика". — Національний університет «Києво-Могилянська академія», Київ, 2024.

Ця дисертація зосереджена на вивченні зв'язків у даних за допомогою застосування штучних нейронних мереж. Ці зв'язки можуть бути представлені в різних формах, і моделюватись по-різному. Їх правильне моделювання є ключовим для успішного вирішення різноманітних завдань, таких як класифікація, регресія та генеративне моделювання.

У сучасних нейронних мережах широко використовуються стандартні метрики для оцінки їх продуктивності, наприклад, класифікаційна точність, середньоквадратична похибка тощо. Проте, високі показники цих метрик не гарантують відсутності помилок або вразливостей у моделях. Моделі можуть видавати помилкові результати з високим рівнем впевненості, особливо при взаємодії з адверсаріальними прикладами — спеціально створеними вхідними даними, які вводять модель в оману.

Це дослідження стосується цієї важливої проблеми шляхом детального вивчення кількісної оцінки невизначеності та стійкості нейронних мереж до адверсаріальних атак. Використовуючи адверсаріальні дані як інструмент, ця робота спрямована на поглиблення розуміння надійності моделей та розроблення більш стійких систем на основі нейронних мереж, які можуть протистояти різноманітним атакам та забезпечувати стабільну продуктивність у

реальних застосуваннях.

Досліджуючи адверсаріальні взаємозв'язки та патерни в даних, ця робота має на меті використовувати їх як метрику генералізації для виявлення слабких місць моделей та оцінки їх здатності до узагальнення. Розуміння того, як моделі реагують на суперечливі збурення, відкриває унікальний погляд на їх внутрішню структуру та механізми прийняття рішень. Це дозволяє не лише виявляти вразливі місця, але й розробляти методи для їх усунення, що підвищує загальну надійність та ефективність моделей.

У рамках цього дослідження вивчаються різні параметризації нейронних мереж для моделювання послідовностей та їх вплив на продуктивність моделей і стійкість до адверсаріальних атак. Особлива увага приділяється новим архітектурам та активаційним функціям, які можуть покращити здатність моделей до генералізації та їхню стійкість. Адверсаріальна стійкість розглядається як важлива метрика для виявлення слабких місць моделей та оцінки їх загальної ефективності.

Дослідження охоплює ефективні параметризації для різних типів вхідних даних, включаючи зображення, мовні сигнали та текст. Застосовуються ці параметризації до різних завдань машинного навчання, таких як класифікація зображень, моделювання мови та регресія на основі латентних дифузійних моделей. Проведені експерименти спрямовані на виявлення того, як різні стратегії параметризації можуть покращити продуктивність моделей, зберігаючи або навіть підвищуючи їх стійкість до адверсаріальних атак.

Отримані результати надають важливі знання для розробки більш надійних та здатних до генералізації моделей машинного навчання. Це сприяє прогресу у цій галузі шляхом виявлення оптимальних технік параметризації, які збалансовують продуктивність та стійкість, та можуть бути застосовані у широкому спектрі практичних задач.

Дисертація складається з кількох розділів, кожен з яких охоплює ключові аспекти дослідження.

Перший розділ, “Геометричні властивості адверсаріальних прикладів”, надає глибоке визначення адверсаріальним атакам, класифікує їх за різними типами та досліджує їх геометричні властивості. Тут розглядаються різні методи створення адверсаріальних прикладів, такі як атаки за градієнтами, методи з обмеженням норми збурення та інші. Аналізуються механізми, за допомогою яких адверсаріальні атаки експлуатують вразливості моделей, та як ці вразливості пов’язані з геометрією простору ознак.

Наступний розділ, “Моделювання сигналів за допомогою механізму уваги з ковзним середнім”, зосереджується на оцінці модифікованої функції уваги для ефективного моделювання послідовностей. Механізм уваги з ковзним середнім пропонується як альтернатива традиційним методам, таким як рекурентні нейронні мережі та стандартні механізми уваги. Розділ детально описує методологію, математичний апарат та алгоритмічну реалізацію запропонованого підходу. Проводиться оцінка його ефективності у завданнях моделювання мовленнєвих сигналів.

Крім того, дисертація містить розділ “Аналіз дифузійного моделювання на прикладі аудіо сигналів”, у якому досліджується використання латентних дифузійних моделей для синтезу аудіо. Розглянуто методи компресії ознак за допомогою маскованих та варіаційних автокодувальників, а також адаптацію компонентів попередньо навченої моделі AudioLDM2 для покращення генерації мовлення. Проведено оцінювання запропонованої моделі за метриками схожості голосу, точності класифікації емоцій та похибок розпізнавання мовлення, що дозволило виявити її переваги та напрямки для подальшого вдосконалення.

Далі йде розділ “Багатовимірні активаційні функції”, який досліджує нові види активаційних функцій, що моделюють взаємозв’язки між багатьма нейронами одночасно. Традиційні активаційні функції зазвичай діють на рівні окремого нейрона, але запропоновані багатовимірні функції дозволяють моделювати складніші залежності та взаємодії у нейронній мережі. Розглянуто

декілька видів таких функцій, їх математичні властивості та вплив на навчання моделі. Емпіричні результати демонструють покращення продуктивності у різних завданнях машинного навчання, включаючи класифікацію, регресію та генеративні моделі.

Розділ “Адверсаріальна стійкість” надає результати експериментів, що оцінюють стійкість розглянутих параметризацій до різних типів адверсаріальних атак. Тут проводиться порівняльний аналіз моделей з різними параметризаціями щодо їх здатності протистояти атакам, таким як PGD (Projected Gradient Descent) та інші. Надано уявлення про те, як різні стратегії параметризації та архітектурні рішення впливають на стійкість моделей до адверсаріальних атак, а також розглянуто методи для покращення цієї стійкості, такі як регуляризація, згладжування міток (англ. label smoothing) та змагальне навчання (англ. adversarial training).

У розділ “Висновки” представлені загальні результати дисертації, підсумовано ключові висновки та їх вплив на сферу машинного навчання. Обговорено значення отриманих результатів для практичного застосування, а також запропоновано потенційні напрямки для майбутніх досліджень. Зокрема, обговорюється можливість подальшого розвитку багатовимірних активаційних функцій, дослідження нових механізмів уваги та глибше вивчення геометричних аспектів адверсаріальної стійкості. Проведені дослідження підтверджують ефективність використання розглянутих нейронних мережевих моделей для підвищення точності класифікації та демонструють складності, які виникають при адверсаріальному тренуванні.

Загалом, ця дисертація робить вагомий внесок у розуміння та покращення стійкості нейронних мереж до адверсаріальних атак, пропонуючи нові підходи до параметризації та моделювання, які можуть бути застосовані у різних сферах машинного навчання. Результати цього дослідження можуть стати основою для розробки більш надійних та ефективних моделей, здатних забезпечувати високу продуктивність та безпеку у реальних застосуваннях.

Проведені експерименти підтверджують, що використання розглянутих параметризацій може підвищити точність класифікації, але також виявляють складності, пов'язані з їх адверсаріальним тренуванням. Подальші дослідження у цьому напрямку можуть призвести до створення моделей, які не лише демонструють високу продуктивність, але й є стійкими до різноманітних атак, що є критично важливим у сучасному світі, де безпека та надійність моделей машинного навчання набувають все більшого значення.

Ключові слова: адверсаріальна стійкість, адверсаріальні приклади, адверсаріальне очищення, механізм уваги, параметризація моделей, дифузійне моделювання, автокодувальники, обробка сигналів, адаптивні алгоритми, алгоритми оптимізації, функції активації, регуляризація нейронні мережі, штучна нейронна мережа, алгоритм, згорткова нейронна мережа, параметри, похибка машинне навчання, класифікація, регресія, генеративне моделювання, комп'ютерний зір, обробка природної мови, аудіо моделювання.

ABSTRACT

Ivaniuk A.O. Study of relationships in data using artificial neural networks. — Qualified research work (manuscript).

Dissertation to obtain the scientific degree of Doctor of Philosophy in the Field of Study 11 “Mathematics and statistics”, Programme Subject Area 113 “Applied mathematics”. — National University of Kyiv-Mohyla Academy, Kyiv, 2024.

This dissertation focuses on studying relationships in data through the application of artificial neural networks. These relationships can be represented in various forms and modeled in different ways. Correct modeling of these relationships is key to successfully solving a variety of tasks, such as classification, regression, and generative modeling.

In modern neural networks, standard metrics are widely used to evaluate their performance, such as classification accuracy, mean squared error, and so on. However, good values of these metrics do not guarantee the absence of errors or vulnerabilities in models. Models can produce erroneous results with a high level of confidence, especially when interacting with adversarial examples—specially crafted input data that mislead the model.

This research addresses this important problem by conducting a detailed study of quantitative assessment of uncertainty and the robustness of neural networks to adversarial attacks. By using adversarial data as a tool, this work aims to deepen the understanding of model reliability and to develop more robust neural network-based systems that can withstand various attacks and provide stable performance in real-world applications.

By investigating adversarial relationships and patterns in data, this work aims to use them as a metric of generalization to identify model weaknesses and assess their ability to generalize. Understanding how models respond to conflicting

perturbations offers a unique perspective on their internal structure and decision-making mechanisms. This allows not only for the identification of vulnerabilities but also for the development of methods to eliminate them, thereby enhancing the overall reliability and efficiency of models.

As part of this research, various parameterizations of neural networks for sequence modeling are studied, as well as their impact on model performance and robustness to adversarial attacks. Special attention is paid to new architectures and activation functions that can improve models' ability to generalize and their robustness. Adversarial robustness is considered an important metric for identifying model weaknesses and evaluating their overall effectiveness.

The research encompasses effective parameterizations for different types of input data, including images, speech signals, and text. These parameterizations are applied to various machine learning tasks, such as image classification, language modeling, and regression based on latent diffusion models. The experiments conducted aim to identify how different parameterization strategies can improve model performance while maintaining or even enhancing their robustness to adversarial attacks.

The results obtained provide important insights for developing more reliable and generalizable machine learning models. This advances the field by identifying optimal parameterization techniques that balance performance and robustness and can be applied in a wide range of practical tasks.

The dissertation consists of several chapters, each covering key aspects of the research.

The first chapter, "Geometric Properties of Adversarial Examples", provides a deep definition of adversarial attacks, classifies them by different types, and explores their geometric properties. Various methods for creating adversarial examples are considered here, such as gradient-based attacks, methods with perturbation norm constraints, and others. The mechanisms by which adversarial attacks exploit model vulnerabilities are analyzed, as well as how these

vulnerabilities are related to the geometry of the feature space.

The next chapter, “Modeling Signals Using the Moving Average Attention Mechanism”, focuses on evaluating a modified attention function for effective sequence modeling. The moving average attention mechanism is proposed as an alternative to traditional methods such as recurrent neural networks and standard attention mechanisms. The chapter provides a detailed description of the methodology, mathematical framework, and algorithmic implementation of the proposed approach. An evaluation of its effectiveness in speech signal modeling tasks is conducted.

Additionally, the dissertation includes the chapter “Analysis of Diffusion Modeling on the Example of Audio Signals”, in which the use of latent diffusion models for audio synthesis is explored. Methods of feature compression using masked and variational autoencoders are considered, as well as the adaptation of components from the pre-trained AudioLDM2 model to improve speech generation. An evaluation of the proposed model was conducted using metrics such as voice similarity, emotion classification accuracy, and speech recognition errors, which allowed for identifying its advantages and directions for further improvement.

Next is the chapter “Multivariate Activation Functions”, which explores new types of activation functions that model interconnections among multiple neurons simultaneously. Traditional activation functions usually operate at the level of individual neurons, but the proposed multivariate functions allow modeling more complex dependencies and interactions in the neural network. Several types of such functions are considered, their mathematical properties, and their impact on model training. Empirical results demonstrate performance improvements in various machine learning tasks, including classification, regression, and generative models.

The chapter “Adversarial Robustness” presents the results of experiments that assess the robustness of the considered parameterizations to various

types of adversarial attacks. A comparative analysis of models with different parameterizations is conducted regarding their ability to resist attacks such as PGD (Projected Gradient Descent) and others. Insights are provided into how different parameterization strategies and architectural decisions affect model robustness to adversarial attacks, and methods for improving this robustness are considered, such as regularization, label smoothing, and adversarial training.

In the chapter “Conclusions”, the general results of the dissertation are presented, key findings are summarized, and their impact on the field of machine learning is discussed. The significance of the obtained results for practical applications is considered, and potential directions for future research are proposed. In particular, the possibility of further development of multidimensional activation functions, exploration of new attention mechanisms, and deeper study of geometric aspects of adversarial robustness are discussed. The conducted studies confirm the effectiveness of using the considered neural network models to improve classification accuracy and demonstrate the complexities that arise during adversarial training.

Overall, this dissertation makes a significant contribution to understanding and improving the robustness of neural networks to adversarial attacks by proposing new approaches to parameterization and modeling that can be applied in various fields of machine learning. The results of this research can serve as a foundation for developing more reliable and efficient models capable of ensuring high performance and security in real-world applications. The experiments conducted confirm that using the considered parameterizations can enhance classification accuracy but also reveal complexities associated with their adversarial training. Further research in this direction may lead to the creation of models that not only demonstrate high performance but are also robust to various attacks, which is critically important in today’s world where the security and reliability of machine learning models are becoming increasingly significant.

Keywords: adversarial robustness, adversarial examples, adversarial

purification, attention mechanism, model parameterization, diffusion modeling, auto-encoders, signal processing, adaptive algorithms, optimization algorithms, activation functions, regularization, neural networks, artificial neural network, algorithm, convolutional neural network, parameters, error, machine learning, classification, regression, generative modeling, computer vision, natural language processing, audio modeling.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. A. Ivaniuk and G. Kriukova, ”On Geometric Properties of Adversarial Examples,”2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, 2021, pp. 964-967, doi: 10.1109/IDAACS53288.2021.9660991.
2. Ivaniuk, A. 2022. Мовне моделювання аудіо з допомогою механізму уваги з рухомим середнім. Могилянський математичний журнал. 5, (Груд 2022), 53–56. DOI:https://doi.org/10.18523/2617-70805202253-56.
3. A. Ivaniuk (2024). ”Latent diffusion model for speech signal processing.”Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems, vol. 61, pp. 43-51, 2024. https://doi.org/10.26565/2304-6201-2024-62-05

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Національний технічний університет України “Київський політехнічний інститут ім. Ігоря Сікорського“, 16 квітня 2021 Геометричні властивості згенерованих adversarial прикладів / секція алгебри, дискретної математики, теорії алгоритмів, інформатики
2. A. Ivaniuk, O. Kravchuk, G. Kriukova (2023). On applications of expanders in manifold regularization. The 14th Ukraine Algebra Conference (p. 70). Sumy, Ukraine: Sumy State Pedagogical University.

ЗМІСТ

ВСТУП

Обґрунтування вибору теми дослідження. Тема дослідження, присвячена різним параметризаціям для ефективного нейромережевого моделювання послідовностей та аспектам їхньої стійкості до адверсаріальних атак, є надзвичайно актуальною в сучасній науці. Зростання обсягів даних та необхідність їх обробки в реальному часі вимагають вдосконалення алгоритмів та моделей для роботи з послідовними даними. Це включає текстові, аудіо- та відеопотоки, які широко використовуються в різних галузях, таких як розпізнавання мовлення, машинний переклад, аналіз природної мови та інші.

Ефективне моделювання послідовностей вимагає розробки нових параметризацій і нейромережевих блоків, що дозволяють підвищити точність та швидкість обробки даних. Використання різних параметризацій, таких як інтерполяційні функції активації чи змішані гаусові моделі (англ. Gaussian Mixture Models), може значно покращити продуктивність моделей. Вибір оптимальної параметризації дозволяє моделі більш точно захоплювати взаємозв'язки в даних, що є критичним для завдань, пов'язаних з обробкою природної мови та іншими послідовними даними.

Стійкість до адверсаріальних атак є ключовим аспектом для будь-якої моделі, яка використовується в реальному середовищі. Адверсаріальні атаки можуть значно знизити точність та надійність моделей, що в свою чергу може призвести до серйозних помилок у застосуваннях. Дослідження методів підвищення стійкості моделей до таких атак, включаючи адверсаріальне тренування, адверсаріальне очищення, є необхідним для забезпечення безпеки та надійності систем, що використовують машинне навчання. Поєднання досліджень у галузі параметризації для ефективного моделювання послідовностей та стійкості до адверсаріальних атак дозволяє створити більш надійні

та ефективні моделі, які можуть бути використані в широкому спектрі завдань. Це дослідження має потенціал для значного внеску в розвиток методів машинного навчання та їх застосування у різних галузях.

Таким чином, вибір теми дослідження обґрунтований необхідністю розробки нових методів для покращення ефективності та надійності моделей машинного навчання, що працюють з послідовними даними.

Мета і завдання дослідження. Метою цього дослідження є розробка та оцінка нових параметризацій для ефективного нейромережевого моделювання послідовностей, а також вивчення їх стійкості до адверсаріальних атак. Зокрема, дослідження спрямоване на підвищення продуктивності моделей при збереженні їх стійкості до адверсаріальних впливів. Це дозволить створити більш надійні та ефективні моделі, які можуть бути використані в широкому спектрі завдань машинного навчання. *Об'єктом* дослідження є нейромережеві моделі для обробки послідовних даних. *Предметом* дослідження є нові методи та підходи до параметризації нейронних мереж, які дозволяють покращити їхню ефективність і стійкість при обробці послідовних даних, включаючи розробку багатовимірних активаційних функцій, дослідження адверсаріальних прикладів та впровадження інноваційних механізмів уваги для моделювання мовлення.

Основні завдання дослідження

1. **Дослідження адверсаріальних прикладів:**
   - Розробити методи для виявлення адверсаріальних прикладів на основі геометричних властивостей латентного простору.
   - Оцінити ефективність автоенкодерів у створенні латентних просторів для розпізнавання адверсаріальних прикладів.
   - Побудувати модель бінарної класифікації для розпізнавання адверсаріальних прикладів.
2. **Моделювання мовлення з використанням механізмів уваги:**

- Вивчити вплив механізму уваги, оснащеного ковзним середнім, на якість синтезу мовлення.
- Порівняти ефективність стандартного механізму уваги та механізму з ковзним середнім у авторегресійних моделях.
- Оцінити поліпшення якості мовлення завдяки впровадженню нових механізмів уваги.

3. **Використання дифузійних моделей для обробки мовних сигналів:**
   - Оцінити ефективність дифузійних моделей у порівнянні з традиційними авторегресійними моделями для задачі синтезу мовлення.
   - Вивчити можливості дифузійних моделей для покращення якості та природності синтезованого мовлення.
   - Провести ретельне порівняння результатів, щоб визначити переваги та недоліки дифузійних моделей.
4. **Розробка та оцінка багатовимірних активаційних функцій:**
   - Розробити нові багатовимірні активаційні функції та оцінити їх вплив на продуктивність нейронних мереж.
   - Провести експерименти з використанням багатовимірних активаційних функцій на різних завданнях, таких як моделювання мови, класифікація зображень та регресія.
   - Оцінити переваги нових активаційних функцій для підвищення продуктивності моделей.
5. **Оцінка адверсаріальної стійкості розглянутих моделей:**
   - Провести експеримент для вимірювання адверсаріальної стійкості класифікаторів зображень з використанням багатовимірних активаційних функцій та механізму уваги з ковзним середнім.
   - Вивчити, як запропоновані параметризації можуть підвищити стійкість моделей до адверсаріальних атак.
   - Оцінити продуктивність моделей, використовуючи відомі мето-

ди протидії адверсаріальним атакам: адверсаріальне тренування та очищення

Таким чином, основна мета дослідження полягає у розвитку нових методів машинного навчання, які забезпечують підвищену надійність та ефективність моделей, сприяючи їх більш широкому та успішному застосуванню у різних сферах.

Методи дослідження. У цьому дослідженні використовувались абляційні дослідження (англ. ablation studies) для оцінки різних параметризацій та блоків для нейронної мережі. Було проведено емпіричний аналіз з використанням різноманітних об'єктивних метрик, відповідних до конкретних задач машинного навчання. Це дозволило комплексно оцінити ефективність запропонованих методів і їх вплив на продуктивність та стійкість моделей до адверсаріальних атак.

Наукова новизна отриманих результатів. Наукова новизна отриманих результатів полягає у впровадженні та дослідженні нових параметризацій для моделей послідовностей, що сприяють підвищенню їх ефективності та стійкості до адверсаріальних атак. Запропоновані методи були апробовані на різноманітних задачах машинного навчання, включаючи класифікацію зображень, моделювання мови та регресію. Виявлено, що нові параметризації, такі як мультиваріативні активаційні функції та різні варіації механізмів уваги, значно покращують продуктивність моделей та їх здатність протистояти адверсаріальним атакам.

Практичне значення отриманих результатів. Отримані результати мають важливе практичне значення в контексті сучасних досліджень машинного навчання. Враховуючи теорію, що стверджує, що «висока перенавченість (англійською overfitting) може погіршити стійку узагальненість», а також гіпотезу про компроміс між точністю та адверсаріальною стійкістю [1], нові техніки, такі як механізм уваги з рухомим середнім(англ. Moving Average

Equipped Gated Attention) [2] та запропоновані багатовимірні активаційні функції, дозволяють нейронним мережам моделювати більш складні закономірность, оскільки вони додають додаткові параметри, що навчаються. Таким чином, у даній роботі проводиться оцінка того, як ці нові техніки впливають на якість моделі та її стійкість до адверсаріальних атак, та перевіряється, чи підтверджується припущення про цей компроміс.

Результати дослідження показують, що нові активаційні функції та механізми уваги можуть суттєво підвищити продуктивність моделей у різних задачах, включаючи моделювання мови, аудіо сигналів, зображень для задач класифікації та регресії. Проте, зважаючи на те, що додаткові параметри можуть збільшувати перенавченість, дослідження також зосереджується на тому, як ці покращення впливають на стійкість моделей до адверсаріальних атак. Цей підхід дозволяє зробити важливі висновки про баланс між складністю моделі та її здатністю до узагальнення на нових даних, що є критичним для розробки надійних та ефективних систем машинного навчання.

Практична значущість отриманих результатів полягає в їхньому потенціалі для широкого впровадження у різні сфери застосування штучного інтелекту. Висновки дослідження можуть бути використані для створення більш стійких до атак і водночас ефективних моделей ML, що мають підвищену здатність до узагальнення. Це є ключовим для критично важливих систем, таких як автономне водіння, медичні системи та системи безпеки, де надійність і стійкість до атак мають вирішальне значення.

Особистий внесок здобувача. Результати, що виносяться на захист, отримані автором самостійно.

Апробація матеріалів дисертації. Основні результати дослідження доповідалися на наукових конференціях різного рівня. Це такі конференції:

- Десята всеукраїнська наукова конференція молодих математиків, Київ, 16–17 квітня 2021 р., онлайн;

- The 11th IEEE International conference on Intelligent Data Acquisition and Advanced computing systems: Technology and Applications, 22-25 вересня, 2021 р, онлайн, секційна доповідь;
- 14 Українська конференція алгебри, Сумський державний педагогічний університет ім. А.С. Макаренка, КНУ ім. Тараса Шевченка, 3-7 липня 2023 р, онлайн;

Структура та обсяг дисертації. Дисертаційна робота складаєтсья зі вступу, п’ятьох розділів, загальних висновків, списку використаних джерел та одного додатку. Обсяг загального тексту дисертації складає 166 сторінок (6.9 д.а.), з них основного тексту 125 (5.3 д.а.). Робота ілюстрована 13 таблицями. Список використаних джерел містить 169 цитувань.

РОЗДІЛ 1
ГЕОМЕТРИЧНІ ВЛАСТИВОСТІ ЗГЕНЕРОВАНИХ ЗМАГАЛЬНИХ ПРИКЛАДІВ

Адверсаріальні(змагальні) приклади — це спеціальні вхідні дані для моделі машинного навчання, які були розроблені, щоб змусити модель зробити неправильний прогноз. Існуючі методи захисту, такі як змагальне тренування, часто вимагають попередніх знань про атаки противника або модифікації цільової моделі. Однак виявлення такого зловмисного введення залишається відкритою проблемою.

У цій роботі розглядаються деякі типи атак білої скриньки (англ. white-box), такі як алгоритм обмеженої пам'яті Бройдена-Флетчера-Голдфарба-Шенно (англ. L-BFGS) [3], знак швидкого градієнта (англ. Fast Gradient Sign Method, FGSM) [4], карта Saliency на основі Якобіана (англ. Jacobian-based Saliency Map Attack, JSMA) [5], DeepFool [3], one-pixel [4] і adversarial patch [6] та досліджуються їхні геометричні властивості. Приипускається, що дані великої розмірності, які подаються на вхід у модель, знаходяться в гладкому різноманітті меншої розмірності (англ. Manifold hypothesis). Шляхом проекції даних у низьковимірний просторі вивчаються геометричні властивості, такі як середні відстані точок даних до центроїдів найближчих сусідів, використовуючи різні метрики відстані [7].

На основі обчислених властивостей було створено модель класифікатора, яка може розрізняти дані, відібрані з вихідного розподілу, і адверсаріальні вхідні дані.

1.1. Визначення і класифікація адверсаріальних атак

Адверсаріальні атаки - це процеси створення адверсаріального прикладу на основі даного натурального зразка і моделі-жертви. На Рисунку 1.1 проілюстровано цей процес створення адверсаріальних прикладів. Позначимо оригінальне зображення x_0, і глибока нейронна мережа (англ. Deep Neural Network, DNN) може правильно передбачити його клас y_0. Мета адверсаріальної атаки - знайти мале збурення $\boldsymbol{\delta}$ так, щоб адверсаріальний приклад $x^* = x_0 + \boldsymbol{\delta}$, який виглядає подібним до x_0 для людей, був невірно класифікований моделлю-жертвою.

Було введено безліч методів атак для створення адверсаріальних прикладів для атак на різні DNN. Загалом, існує два види цілей атак: таргетовані та нетаргетовані. Для нетаргетованої атаки атака вважається успішною, якщо приклад класифікований з будь-яким неправильним класом. Візьмемо для прикладу Рисунок 1.1. Як тільки зображення панди не класифікується як панда, атака успішна. Для таргетованої атаки атака вважається успішною, тільки коли адверсаріальний приклад класифікується як цільовий клас. У цьому прикладі, якщо цільовий клас - ’гібон’, атака вважається успішною, тільки коли праве зображення помічене як ’гібон’ класифікатором-жертвою.

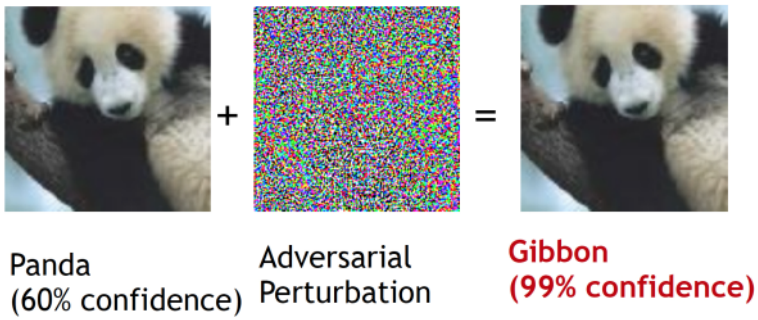


Рис. 1.1: Приклад [4] створення адверсаріального зображення

На основі необхідної інформації методи атак було запропоновано [8] поділити на три категорії: (1) на основі градієнта, (2) на основі оцінки, та (3) на основі рішень. Більшість цих методів можуть виконувати як таргетовані, так

і нетаргетовані атаки. Загалом, кожний конкретний метод атаки належить до однієї з трьох категорій, хоча нещодавно було показано, що ансамбль атак з кількох категорій може потенційно призвести до сильнішої атаки [9]. Якщо вся інформація про модель-жертву, така як структура моделі, параметри тощо, розкрита нападнику, сценарій називається білою коробкою (англ. white-box). Якщо доступні лише передбачені оцінки, сценарій називається чорною коробкою з м'якими мітками (англ. black-box with soft labels). Якщо розкриті лише передбачені класи, сценарій називається чорною коробкою з жорсткими мітками(англ. black-box with hard labels). Існують також сірі коробки, де частина інформації про модель доступна. Зазначимо, що частина інформації про модель, яка є доступною/недоступною, залежить від проблеми. Наприклад, [10] визначили сіру коробку (англ. grey-box) як сценарій, де нападник має доступ до класифікатора, але не до дизайну детектора адверсаріальних прикладів.

1.1.1. Атака на основі градієнта

Багато існуючих методів атак потрапляють у цю категорію. Ці методи використовують градієнти loss-функції відносно вхідного зображення для формування адверсаріальних прикладів. Наприклад, метод FGSM генерує адверсаріальні приклади на основі знаку градієнтів і використовує розмір кроку для контролю ℓ_∞ норми збурення. Основний Ітеративний Метод (англ. Basic Iterative Metho, BIM) [6] та Атака Проектованого Градієнтного Спуску (англ. Projected Gradient Descent? PGD) [11] можна розглядати як ітераційні версії FGSM. PGD створює адверсаріальні приклади, виконуючи FGSM для фіксованої кількості ітерацій або до досягнення невірної класифікації. Найбільш ефективні методи атак на основі градієнта на сьогодні - це C&W [12] та PGD. Атаки C&W і PGD часто використовуються для оцінки алгоритмів захисту через їхню ефективність [13]. Оскільки для виконання атаки потрібна інформація про градієнт, атака на основі градієнта в основному призначена для

сценарію білої коробки. Процес створення адверсаріальних прикладів можна сформулювати як задачу оптимізації. Залежно від формулювань оптимізації, методи атак на основі градієнта можна додатково розділити на дві підкатегорії.

Методи на основі формулювання задачі оптимізації з обмеженнями: Перша підкатегорія складається з методів на основі формулювання задачі оптимізації з обмеженнями. Враховуючи модель з фіксованим параметром Θ і вхідною парою (x_0, y_0), процес створення адверсаріальних прикладів x^* можна описати наступною задачею оптимізації:

$$x^* = x_0 + \boldsymbol{\delta} \quad \text{з} \quad \boldsymbol{\delta} = \arg\max_{\boldsymbol{\delta} \in \mathcal{S}} L(\theta, x_0 + \boldsymbol{\delta}, y), \tag{1.1.1}$$

де L є loss-функцією, $\boldsymbol{\delta}$ є адверсаріальним шумом, а $\mathcal{S} \in R^d$ є множиною допустимих шумів, зазвичай обирається $\mathcal{S} = \{\|\boldsymbol{\delta}\|_\infty \leqslant \epsilon\}$ для деякого ϵ.

Мета оптимізації - знайти адверсаріальне збурення, яке призводить до неправильної класифікації. Тому для нетаргетованої атаки loss-функція L може бути функцією втрат, що використовується для навчання класифікатора, тоді як для таргетованої атаки слід використовувати інші loss-функції, призначені для цільового класу. Максимізуючи (1.1.1), нападник змушує класифікатор робити помилку в задачі класифікації. Щоб гарантувати, що збурення не є надто великим, щоб x_0 і x^* були нерозрізнюваними для людини, пошуковий простір обмежений ϵ-кулею навколо введення.

Задачу оптимізації, визначену в (1.1.1), можна переформулювати як

$$\arg\max_{\delta \in \mathcal{S}} L(\boldsymbol{\theta}, x_0 + \delta, y) = \arg\max_{\|\delta\|_\infty \leqslant \epsilon} L(\boldsymbol{\theta}, x_0 + \delta, y), \tag{1.1.2}$$

яку можна розв'язати методом градієнтного спуску.

Атака проектованого градієнтного спуску (PGD) створює адверсаріальні приклади, розв'язуючи цю задачу оптимізації за допомогою методу проектованого градієнтного спуску. Її запропонували [11], які знаходять адверсаріальні приклади в ϵ-кулі введення. Атака PGD оновлюється в на-

прямку, який найбільше знижує ймовірність початкового класу, а потім проектує результат назад в ϵ-кулю введення. Рівняння оновлення для атаки PGD виглядає так:

$$x^{t+1} = \Pi_\epsilon \left\{ x^t + \alpha \cdot \text{sign}\Big(\nabla_x L(\boldsymbol{\theta}, x^t, y)\Big), x_0 \right\}, \tag{1.1.3}$$

де x_0 є початковим входом, x^t - оновлені вхідні на кроці t, ϵ контролює максимальне спотворення, α є розміром кроку, а $\nabla_x L(\boldsymbol{\theta}, x^t, y)$ представляє градієнт класифікаційної loss-функції $L(\cdot)$ відносно входу x^t.

Зазвичай атака PGD створює адверсаріальний приклад шляхом додавання або віднімання невеликого помилкового члена, α, до кожного виміру вхідних даних. Рішення про додавання або віднімання помилкового члена залежить від того, чи є знак градієнта для виміру введення позитивним або негативним. Потім результат попереднього кроку проектується на ϵ-кулю навколо початкового введення. Якщо (1.1.3) виконує тільки один крок, це еквівалентно методу FGSM. Якщо виконується кілька кроків, це атака PGD або BIM.

Автоматичний варіант PGD. Алгоритм PGD має три слабкі місця в стандартному формулюванні та використанні в контексті адверсаріальної стійкості. По-перше, фіксований розмір кроку є неоптимальним, оскільки навіть для опуклих задач оптимізації це не гарантує збіжності, а продуктивність алгоритму сильно залежить від вибору його значення, див. напр. [14]. По-друге, загальна схема загалом не враховує бюджет, виділений на атаку: як було показано, значення фуннкції втрат стабілізується після декількох ітерацій, за винятком надзвичайно малих розмірів кроку, які, однак, не призводять до кращих результатів. Відповідно, оцінка сили атаки за кількістю ітерацій може дати оманливий результат [15]. Нарешті, алгоритм не враховує тенденцію, тобто не розглядає, чи оптимізація розвивається успішно і не здатний на це реагувати [9].

Дана модифікація, під назвою Auto PGD (APGD) [9] була створеня для того щоб усунути ці проблеми. Основна ідея полягає в розділі наявних n іте-

рацій на початкову фазу дослідження, де здійснюється пошук вдалих початкових точок у допустимій множині, і фазу експлуатації, під час якої відбувається ”максимальне використовувати накопичених знань”. Перехід між цими фазами керується поступовим зменшенням розміру кроку. Великий розмір кроку дозволяє швидко переміщатися в $\mathcal{S}$, тоді як менший розмір кроку більш ефективно максимізує цільову функцію локально. Однак зменшення розміру кроку не заплановано заздалегідь, а керується тенденцією оптимізації: якщо значення цільової функції зростає досить швидко, то розмір кроку, швидше за все, правильний, в іншому випадку доцільно його зменшити.

Хоча крок оновлення є стандартним, що відрізняє цей алгоритм від звичайного PGD, це вибір розміру кроку протягом ітерацій, який адаптовано до загального бюджету і до прогресу оптимізації. Якщо розмір кроку зменшується, максимізація перезапускається з найкращої точки, знайденої до цього моменту. Крок градієнта: Оновлення в APGD тісно слідує класичному алгоритму та лише додає термін імпульсу. Нехай ηk буде розміром кроку на ітерації k, тоді крок оновлення виглядає так:

$$\begin{aligned} zk+1 &= P_{\mathcal{S}}\left(xk+\eta k\nabla f(xk)\right) \\ xk+1 &= P_{\mathcal{S}}\left(xk+\alpha\cdot(zk+1-xk)\right. \\ &\qquad \left.+(1-\alpha)\cdot(xk-xk-1)\right), \end{aligned} \tag{1.1.4}$$

де $\alpha \in [0,1]$ (зазвичай $\alpha = 0.75$) регулює вплив попереднього оновлення на поточне. Оскільки на початкових ітераціях APGD розмір кроку особливо великий, алгоритм мінімізує вплив попередніх ітерацій.

Вибір розміру кроку: Алгоритм починається з розміру кроку $\eta 0$ на першій ітерації (також $\eta 0$ фіксується: $\eta 0 = 2\epsilon$), і з огляду на бюджет ітерацій, визначаються контрольні точки $w_0 = 0, w_1, \ldots, w_n$, на яких алгоритм вирішує, чи потрібно зменшити поточний розмір кроку вдвічі. Алгоритм передбачає дві умови:

1. $\sum\limits_{i=w_{j-1}}^{w_j-1} \mathbf{1}_{f(xi+1)>f(xi)} < \rho\cdot(w_j-w_{j-1})$,

2. $\eta w_{j-1} \equiv \eta w_j$ і $f_{\max} w_{j-1} \equiv f_{\max} w_j$,

де $f_{\max} k$ є найвищим значенням цільової функції, знайденим за перші k ітерацій. Якщо одна з умов виконується, то розмір кроку на ітерації $k = w_j$ зменшується вдвічі, і $\eta k \eta w_j / 2$ для кожного $k = w_j + 1, \ldots, w_{j+1}$.

Умова 1: підраховує, в скількох випадках з останньої контрольної точки w_{j-1} крок оновлення був успішним у збільшенні f. Якщо це сталося щонайменше для частки ρ від загальної кількості кроків оновлення, то розмір кроку залишається незмінним, оскільки оптимізація йде належним чином ($\rho = 0.75$).

Умова 2: виконується, якщо розмір кроку не було зменшено на останній контрольній точці *i* не було покращення у значенні цільової функції з останньої контрольної точки. Це запобігає застряганню у потенційних циклах.

Перезапуск з найкращої точки: Якщо на контрольній точці w_j розмір кроку зменшується вдвічі, то встановлюється $x w_j + 1 x_{\max}$, тобто алгоритм перезапускається з точки, яка досягла найвищого значення цільової функції $f_{\max}$ до цього моменту. Це мотивується тим, що зменшення η веде до більш локалізованого пошуку, і це слід робити в околицях поточного найкращого кандидата на рішення.

Відкриття та експлуатація: в ідеалі, алгоритм поступово повинен переходити від дослідження всієї допустимої множини $\mathcal{S}$ до локальної оптимізації. Цей перехід регулюється поступовим зменшенням розміру кроку та вибором, коли його зменшувати, тобто контрольними точками w_j. Також, алгоритм задумувався так, щоб дозволити відносно довгу початкову фазу дослідження, а потім, можливо, частіше оновлювати розмір кроку, переходячи до експлуатації. Насправді, з меншими розмірами кроків поліпшення цільової функції ймовірно частіші, але також меншого розміру, тоді як важливість використання всього простору входів підтверджується успіхом випадкових перезапусків у звичайній PGD атаці. Контрольні точки фіксуються як $w_j = \lceil p_j \rceil \leqslant$, з

$p_j \in [0, 1]$, визначених як $p_0 = 0$, $p_1 = 0.22$ та

$$p_{j+1} = p_j + \max\{p_j - p_{j-1} - 0.03, 0.06\}.$$

Важливо також те, що довжина періоду $p_{j+1} - p_j$ зменшується на кожному кроці на 0.03, але вони мають принаймні мінімальну довжину 0.06.

Хоча запропонована схема має декілька параметрів, які можуть бути налаштовані, вони фіксуються на вказаних значеннях так, що єдиною вільною змінною є бюджет: N_{iter}.

Методи на основі формулювання задачі оптимізації з регуляризацією: Друга підкатегорія складається з методів на основі формулювання задачі оптимізації з регуляризацією. Атака C&W [16], є репрезентативною і, безперечно, однією з найсильніших методів атак. Вона може виконувати як таргетовані, так і нетаргетовані атаки, які формулюються як наступна задача оптимізації:

$$x^* = \arg\min_x \left\{ \|x - x_0\|_2^2 + c \cdot l(x) \right\}, \tag{1.1.5}$$

де перший член забезпечує мале спотворення початкового входу, а другий член є loss-функцією, яка визначена наступним чином:

$$l(x) = max(max\{f(x)_i : i \neq y_0\} - f(x)_{y_0}, -k) \tag{1.1.6}$$

яка вимірює успіх атаки, а y_0 позначає істинний клас входу x_0, $f(x)_{y_0}$ представляє оцінку передбачення входу x з класом y_0. Параметр $c > 0$ контролює компроміс між спотворенням та успіхом атаки.

Для нетаргетованої атаки, коли нападник хоче лише, щоб класифікатор помилився і не переймається передбаченою міткою адверсаріального прикладу, $g(x)$ визначається як:

$$g(x) = \max\{f(x)_{y_0} - \max_{i \neq y_0} f(x)_i, 0\},$$

де $f(x)_i$ представляє оцінку передбачення входу x з класом i. Мінімізація $g(x)$ зробить оцінку передбачення істинного класу y_0 меншою, ніж оцінки передбачення інших класів. Тому адверсаріальний приклад x^* буде класифіковано в неправильний клас.

Якщо нападник хоче, щоб адверсаріальний приклад був класифікований у конкретний цільовий клас з міткою t, де $t \neq y_0$, тоді $g(x)$ визначається як:

$$g(x) = \max\{\max_{i \neq t} f(x)_i - f(x)_t, 0\}.$$

Наведена вище loss-функція таргетованої атаки підвищить ймовірність передбачення цільового класу t до більшої, ніж оцінки інших класів. [16] показали, що їхня атака може успішно обійти десять різних методів захисту, призначених для виявлення адверсаріальних прикладів.

Натхненні методом elastic-net, запропонованим [17], [18] запропонували атаку з еластичною сіткою (EAD), яка також належить до цієї підкатегорії методів на основі формулювання задачі оптимізації з регуляризацією. EAD формулює процес створення адверсаріальних прикладів як задачу оптимізації з регуляризацією еластичною сіткою, що можна розглядати як розширену версію атаки C&W. Їхнє формулювання атак з еластичною сіткою на класифікатор для створення адверсаріального прикладу щодо зразка з класом виглядає наступним чином:

$$x^* =_x \left\{ \|x - x_0\|_2^2 + \beta \|x - x_0\|_1 + c \cdot l(x) \right\},$$

де $l(x)$ той самий, що й у атаці C&W. Автори показали, що EAD може покращити переносимість атаки та доповнити адверсаріальне навчання.

Швидкий знак градієнта

Було висунуто гіпотезу, що нейронні мережі вразливі до лінійних збурень вхідних даних, що мотивує алгоритм швидкого знаку градієнта, який був розроблений для швидкої роботи [4]. Він оновлює оригінальне зображення за

допомогою наступної формули:

$$x' = x - \varepsilon \cdot \text{sign}(\nabla loss_{f,x})$$

де ε — це невелике число, а $\nabla loss_{f,x}$ — це градієнт функції втрат по відношенню до вхідних даних x і класифікатора f [4].

Цей алгоритм було вдосконалено у [6], де було запропоновано робити багато маленьких кроків α замість одного великого кроку розміром ε в напрямку знаку градієнта.

JSMA Атака на основі карти значущості Якобіана (англ. Jacobian-based Saliency Map Attack, або JSMA) є жадібним алгоритмом, який ітеративно вибирає пікселі, що мають найбільший вплив на цільовий результат класифікації [5]. Для цього необхідно визначити поняття карти значущості:

$$\alpha_{pq} = \sum_{i \in \{p,q\}} \frac{\delta C(x)_t}{\delta x_i},$$

$$\beta_{pq} = \left(\sum_{i \in \{p,q\}} \sum_{j} \frac{\delta C(x)_j}{\delta x_i} \right) - \alpha_{pq},$$

де α_{pq} позначає, наскільки зміна обох пікселів p і q вплине на цільовий результат класифікації t, а β_{pq} показує, наскільки зміна p і q змінить всі інші виходи. Знаючи α_{pq} і β_{pq} алгоритм вибирає:

$$(p*, q*) = \arg\max_{(p,q)} (-\alpha_{pq} \cdot \beta_{p,q}) \cdot (\alpha_{pq} > 0) \cdot (\beta_{pq} < 0)$$

Deepfool Deepfool є технікою нетаргетованої атаки, оптимізованої для метрики відстані l_2 [19]. Це ефективний метод, який створює ”ближчі”адверсаріальні приклади. Було запропоновано змінювати оригінальне зображення x_0 з мінімальними збуреннями, які змінюють рішення класифікатора. Значення збурення r відповідає ортогональній проекції x_0 на афінну гіперплощину $\Phi = \{x : \omega^T x + b = 0\}$. Воно задається наступною формулою:

$$r = \arg \min_{\text{sign}(f(x_0+r)) \neq \text{sign}(f(x_0))} \|r\|_2 = -\frac{f(x_0)}{\|\omega\|_2^2}\omega,$$

де припускається, що мітка класифікації k дорівнює $\text{sign}(f(x))$ і $\Phi = \{x : f(x) = 0\}$.

Якщо f є диференційованим класифікатором, алгоритм DeepFool пропонує ітеративно лінеаризувати f навколо поточної точки x_i, а мінімальне збурення лінеаризованого класифікатора обчислюється як

$$\arg \min_{r_i : f(x_i) + \nabla f(x_i)^T r_i = 0} \|r_i\|_2$$

Алгоритм зупиняється, коли x_{i+1} змінює знак класифікатора.

Вибір метрики відстані. Формулювання, засновані на оптимізації з обмеженнями, а також задачі оптимізації з регуляризацією, потребують вимірювання відстані для кількісної оцінки різниці між початковим прикладом і збуреним прикладом. Незважаючи на те, що норма ℓ_∞ використовується для атак на основі PGD (1.1.1), а норма ℓ_2 широко використовується в атаках на основі C&W (1.1.5), загалом можна використовувати будь-яку норму ℓ_p для обох формулювань. Наприклад, норма ℓ_1 та суміш різних норм використовувалися в [18]. Більше того, нещодавно було визнано, що атаки на основі норм ℓ_p можуть бути нереалістичними, тому дослідники почали вивчати більш гнучкі множини збурень. Наприклад, [20] використовують відстань Васерштейна для вимірювання сили збурення, а [21] запропонували навчання множин збурень на основі людського сприйняття [8].

Атака з Мінімальним Викривленням (англ. Fast Adaptive Boundary Attack, або FAB)

FAB-атака [22] є методом білого ящика, що генерує зразки, які потребують мінімальних змін вхідних даних для зміни їхньогр класифікаційного результату. Вона працює з різними l_p-нормами (l_1, l_2, l_∞) і забезпечує високу ефективність обчислень.

Визначення мінімальних адверсаріальних збурень

Нехай $f : \mathbb{R}^d \to \mathbb{R}^K$ — класифікатор, що присвоює кожному вхідному $x \in {}^d$ один із K класів. Мінімальна атакація відносно l_p-норми визначається

як:

$$\delta_{\min,p} = \arg\min_{\delta\in^d} \|\delta\|_p, \quad \text{що підлягає} \quad \max_{l\neq c} f_l(x+\delta) \geqslant f_c(x+\delta), \quad x+\delta \in C. \tag{1.1.7}$$

Ця оптимізаційна задача є нелінійною та NP-важкою для нетривіальних класифікаторів.

Проекція на гіперплощину з обмеженнями

Розглянемо гіперплощину $\pi : \langle w, x\rangle + b = 0$, де $w \in {}^d$ — нормальний вектор, а $b \in$ — зміщення. Для проекції точки x на π з обмеженнями в коробці $C = \{z \in {}^d : l_i \leqslant z_i \leqslant u_i\}$ розв'язується задача мінімізації:

$$z^* = \arg\min_{z\in^d} \|z-x\|_p, \quad \text{що підлягає} \quad \langle w, z\rangle + b = 0, \quad l_i \leqslant z_i \leqslant u_i, \quad i = 1, \ldots, d. \tag{1.1.8}$$

Якщо ця задача немає розв'язку (англ. no feasible solution), то використовується наступна альтернатива:

$$z_i' = \begin{cases} l_i, & \text{якщо} \quad \rho w_i > 0, \\ u_i, & \text{якщо} \quad \rho w_i < 0, \\ x_i, & \text{якщо} \quad w_i = 0 \end{cases}, \quad \text{де} \quad \rho = \text{sign}(\langle w, x\rangle + b). \tag{1.1.9}$$

Алгоритм FAB-Атаки

Алгоритм FAB-атаки полягає в ітеративному наближенні мінімальних адверсаріальних прикладів:

1. Ініціалізація: Початковою точкою є вхідний зразок x_{orig}, який правильно класифікується моделлю.

2. Проекція на гіперплощину: - Використовується лінійне наближення класифікатора для обчислення проекції поточної точки на гіперплощину рішень.

3. Крок градієнта з упередженням (агл. Biased gradient step): - Наступна точка обирається як опукла комбінація проекцій поточної точки та оригінального зразка x_{orig}.

$$x^{(i+1)} = (1-\alpha)\text{proj}_p(x^{(i)}, \pi_s, C) + \alpha\text{proj}_p(x_{\text{orig}}, \pi_s, C), \tag{1.1.10}$$

де α обирається як відносна величина між відстанями поточної точки $x^{(i)}$ та оригінального зразка x_{orig} до гіперплощини рішень.

4. Крок екстраполяції: - Для прискорення перетину гіперплощини додається невеликий крок екстраполяції $\eta \geqslant 1$:

$$x^{(i+1)} = \text{proj}_C((1-\alpha)(x^{(i)} + \eta\delta^{(i)}) + \alpha(x_{\text{orig}} + \eta\delta^{(i)}_{\text{orig}})), \tag{1.1.11}$$

де $\delta^{(i)} = \text{proj}_p(x^{(i)}, \pi_s, C) - x^{(i)}$ та $\delta^{(i)}_{\text{orig}} = \text{proj}_p(x_{\text{orig}}, \pi_s, C) - x_{\text{orig}}$.

5. Крок назад: - Після знаходження нового зразка $x^{(i+1)}$, якщо він класифікується неправильно, виконується крок назад до оригінального зразка для зменшення норми викривлення:

$$x^{(i+1)} = (1-\beta)x_{\text{orig}} + \beta x^{(i+1)}, \beta \in (0,1). \tag{1.1.12}$$

6. Фінальний пошук: - Після завершення ітерацій алгоритм виконує фінальний пошук, щоб знайти точку на сегменті $[x_{\text{out}}, x_{\text{orig}}]$, де x_{out} — найближча точка, класифікована неправильно. Це виконується ітеративно, використовуючи формулу:

$$x_{\text{temp}} = x_{\text{out}} - \frac{(f_s(x_{\text{out}}) - f_c(x_{\text{out}}))(x_{\text{out}} - x_{\text{orig}})}{f_s(x_{\text{out}}) - f_c(x_{\text{out}}) + f_s(x_{\text{orig}}) - f_c(x_{\text{orig}})}. \tag{1.1.13}$$

Переваги FAB-Атаки

FAB-атака забезпечує мінімальне викривлення вхідних даних, є ефективною з точки зору обчислень і стійкою до проблем градієнтного маскування, що робить її надійним інструментом для оцінки стійкості моделей.

1.1.2. Score-based атаки

У реальності детальна інформація про модель, така як градієнт, може бути недоступною для нападників. Методи атак на основі оцінки не вимагають доступу до градієнтів. Вони виконують адверсаріальні атаки на основі вихідних оцінок $f(x)_i$ класифікатора-жертви. Наприклад, [23] запропонували метод для оцінки градієнта з інформацією про оцінки та створення адверсаріальних прикладів з оціненим градієнтом. [24] представили метод, який використовує природну еволюційну стратегію для оцінки градієнта та створення адверсаріальних прикладів. Загалом, атаки на основі оцінки можна поділити на дві підкатегорії [8].

Методи на основі оцінки градієнта: Перша підкатегорія - методи на основі оцінки градієнта. Як випливає з назви, ці методи спочатку оцінюють градієнти або знаки градієнтів, а потім створюють адверсаріальні приклади, використовуючи оцінену інформацію.

Атака на основі оптимізації нульового порядку (ZOO), запропонована [23], використовує метод кінцевих різниць для оцінки градієнта loss-функції відносно входу. Потім застосовується атака C&W для створення адверсаріального прикладу. Вона використовує наступну формулу для оцінки градієнта:

$$\frac{\partial f(x)}{\partial x_{(i)}} \approx \frac{f(x+h_i) - f(x-h_i)}{2h},$$

де h - мала константа, а $_i$ - стандартний базисний вектор з лише i-м компонентом, рівним 1, а i варіюється від 1 до розмірності введення. Час, необхідний для оцінки градієнтів, зростає з розмірністю. Коли розмірність введення велика, автори запропонували кілька технік для масштабування оцінки, роблячи можливим створення адверсаріальних прикладів у розумний час для великих DNN, навчених на великих наборах даних, таких як ImageNet [25].

Було запропоновано багато інших методів для ефективної оцінки граді-

єнта та створення адверсаріальних прикладів на його основі, такі як атака NES [24] та атака Bandits [26]. Атака NES використовує природні еволюційні стратегії (отже, акронім NES) для оцінки градієнта loss-функції відносно введення, а потім створює адверсаріальні приклади на основі оціненого градієнта:

$$\frac{1}{\sigma n}\sum_{i=1}^{n} {}_i f(x+\sigma_i),$$

де n - кількість пошуків для оцінки градієнта, ${}_i$ - випадковий напрямок, взятий з $\mathcal{N}(0, I)$, а σ - стандартне відхилення кроку пошуку. Автори також розширили метод для налаштування з частковою інформацією, де доступна лише частина оцінок або топ-k відсортованих міток.

Насправді, найважливіша інформація для адверсаріальної атаки - це знак градієнта, який визначає напрямок оптимізації. Багато методів запропоновані для оцінки знака градієнта та використання цієї інформації для створення адверсаріальних прикладів; помітні приклади включають SignSGD [27] та Sign hunter [28]. Для подальшого підвищення ефективності оцінки градієнта або знака, деякі дослідники пропонують використовувати модель-заміщення, яка є моделлю, навченою на тих самих даних, що й модель-жертва, і працює аналогічно моделі-жертві. Репрезентативні приклади включають Subspace attack [29] та Transfer-based prior [30]. Також, [31] показали, як використовувати розподіл даних для визначення важливих підпросторів для атак чорної скриньки. Ці методи атак також належать до категорії на основі оцінки градієнта, оскільки їх ключовим елементом є ефективна оцінка інформації про градієнт.

Атака Square

Алгоритм Square Attack [32] є ефективною атакою чорного ящика на основі випадкового пошуку. Ця атака створює l_2- та l_∞-адверсаріальні приклади, не покладаючись на локальну градієнтну інформацію, тому підхід з маскуванням градієнту не захистить від неї.

Атака Square базується на випадковому пошуку, який є відомою ітеративною технікою в оптимізації, запропонованою Растрігіним у 1963 році. Головна ідея алгоритму полягає у виборі випадкового оновлення δ на кожній ітерації та додавання цього оновлення до поточної точки $\hat{x}$, якщо це покращує цільову функцію.

Визначення адверсаріальних прикладів для моделі загроз l_p

Нехай $f : [0,1]^d \to \mathbb{R}^K$ — це класифікатор, де d — розмірність вхідних даних, K — кількість класів, а $f_k(x)$ — передбачуваний бал приналежності x до класу k. Класифікатор присвоює клас $\arg\max_{k=1,\dots,K} f_k(x)$ вхідному x. Мета нетаргетованої атаки — змінити правильно передбачений клас y для точки x. Точка $\hat{x}$ називається адверсаріальним прикладом з обмеженням l_p-норми ϵ для x, якщо:

$$\arg\max_{k=1,\dots,K} f_k(\hat{x}) \neq y, \quad \|\hat{x} - x\|_p \leqslant \epsilon, \quad \text{та} \quad \hat{x} \in [0,1]^d, \tag{1.1.14}$$

де $\hat{x}$ є зображенням. Завдання полягає в знаходженні $\hat{x}$ шляхом розв'язання задачі оптимізації з обмеженнями:

$$\min_{\hat{x}\in[0,1]^d} L(f(\hat{x}), y), \quad \text{що підлягає} \quad \|\hat{x} - x\|_p \leqslant \epsilon, \tag{1.1.15}$$

для деякої функції втрат L. У наших експериментах використовується

$$(L(f(\hat{x}), y) = f_y(\hat{x}) - \max_{k\neq y} f_k(\hat{x}) \tag{1.1.16}$$

Атака Square для l_∞-норм

Ініціалізація: Як ініціалізацію можна використати вертикальні смуги шириною один піксель, де колір кожної смуги вибирається рівномірно випадково з множини $\{-\epsilon, \epsilon\}^c$, де c — кількість колірних каналів.

Розподіл вибірки: Розподіл вибірки P для l_∞-норми вибирає рідкі оновлення $\hat{x}$ з $\|\delta\|_0 = h \cdot h \cdot c$, де $\delta \in \{-2\epsilon, 0, 2\epsilon\}^d$, і ненульові елементи форму-

ються в квадрат. Після проекції на l_∞-кулю радіуса ϵ (етап 5 алгоритму) всі компоненти задовольняють $\hat{x}_i \in \{x_i - \epsilon, x_i + \epsilon\}$.

Атака Square для l_2-норм

Ініціалізація: Ініціалізація l_2-збурень полягає у створенні таблиці розміром 5×5, де кожне порушення має форму, описану нижче в розподілі вибірки. Порушення $\hat{x} - x$ масштабується до l_2-норми ϵ, а результат $\hat{x}$ проектується на $[0, 1]^d$.

Розподіл вибірки: Введемо нове оновлення η, яке має два ”центри”з великим абсолютним значенням та протилежними знаками, тоді як інші компоненти мають нижчі абсолютні значення. Це дозволяє локалізовані зміни з високим контрастом між різними половинами, що покращує ефективність запитів. Конкретно, η визначається наступним чином:

$$\eta_{r,s}^{(h_1,h_2)} = \sum_{k=0}^{M(r,s)} \frac{1}{(n+1-k)^2}, \quad n = \left\lfloor \frac{h_1}{2} \right\rfloor, \tag{1.1.17}$$

де $M(r,s) = n - \max\{|r - \lfloor \frac{h_1}{2} \rfloor - 1|, |s - \lfloor \frac{h_2}{2} \rfloor - 1|\}$. Потім η вибирається рівномірно випадково.

Емпіричне та Теоретичне Обґрунтування Square Attack

Дана схема відрізняється від класичного випадкового пошуку тим, що порушення $\hat{x} - x$ конструюються таким чином, щоб на кожній ітерації вони лежали на межі l_∞- або l_2-кулі перед проекцією на область зображення $[0, 1]^d$.

Попередні дослідження авторів цього адгоритму показали, що вона перевершує інші методи в ефективності запитів та досягненні успіху при створенні адверсаріальних прикладів як для l_∞-, так і для l_2-норм.

Інші методи: Друга підкатегорія методів на основі оцінки складається з методів, які не оцінюють інформацію, пов’язану з градієнтом, для створення адверсаріальних прикладів. Наприклад, [33] запропонували Gaussian black-box adversarial attack (Nattack), яка шукає адверсаріальні приклади, моделюючи адверсаріальну популяцію за допомогою гауссового розподілу. Інтуїція

полягає в тому, що можна знайти різні адверсаріальні приклади для одного введення, використовуючи різні методи атак, що свідчить про існування популяції адверсаріальних прикладів. Інші методи, що належать до цієї підкатегорії, включають GenAttack [34], Simple Black-Box Attack [35].

1.1.3. Атака на основі рішень

У багатьох практичних ситуаціях нападник має доступ лише до передбачених класів моделі, але не до будь-якої інформації про градієнт чи оцінки. Коли доступний лише передбачений клас $c(x)$, методи на основі градієнтів та оцінок не працюють. [36] представили метод трансферної атаки, який вимагає лише спостереження за класами, передбаченими моделлю. Основна ідея полягає в тому, щоб навчити модель-заміщення, яка схожа на цільову модель, і шукати адверсаріальні приклади для неї. Boundary Attack була запропонована згодом [37], яка шукає адверсаріальні приклади на основі випадкового блукання на межі класифікаційного порогу рішень. Було запропоновано багато розширень Boundary Attack для покращення її ефективності та продуктивності [38–41]. Існують також методи атак на основі рішень, які не є ані трансферними, ані на основі випадкового блукання [24, 29, 42]. Загалом, мета атаки на основі рішень - створення адверсаріальних прикладів з передбаченими класами, повернутими моделлю-жертвою. Методи атак на основі рішень можна поділити на три підкатегорії [8].

Атаки на основі трансферу: Перша категорія складається з методів трансферної атаки. Різні дослідники спостерігали, що якщо дві нейронні мережі навчені на схожих даних, навіть якщо ці дві моделі мають дуже різні структури, адверсаріальні приклади, створені на одній моделі, можуть бути використані для обману іншої. На основі цього спостереження [36] запропонували навчити модель-заміщення на невеликій кількості навчальних даних і створити адверсаріальні приклади на основі цієї моделі-заміщення. Вони показали, що адверсаріальні приклади, створені на моделі-заміщенні, також

можуть обманути цільовий класифікатор. Запропонований метод не вимагає занадто багато зразків для навчання моделі-заміщення і може досягти відносно високого рівня успішності в нетаргетованих завданнях. Однак метод не працює добре в таргетованих завданнях. [43] запропонували підхід на основі ансамблю для створення трансферних адверсаріальних прикладів. Цей підхід ансамблю збільшує частку таргетованих адверсаріальних прикладів, що переносяться з їхніми цільовими мітками.

Атаки на основі випадкового блукання: Друга категорія складається з методів, що базуються на випадковому блуканні на межі. [37] запропонували Boundary Attack, метод, який не залежить від градієнта loss-функції відносно входу і добре працює в обох умовах: таргетованих і нетаргетованих. При таргетованих атаках, метод починається з зразка, віднесеного до цільового класу, і намагається мінімізувати збурення, залишаючись адверсаріальним. Процес Boundary Attack показано на Рисунку 1.2.

Визначимо простір введення як $\mathcal{D}$, процес вибору $\boldsymbol{\eta}^t$ з запропонованого розподілу виглядає наступним чином:

- Вибірка $\boldsymbol{\eta}^t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
- Нормалізація і обрізка $\boldsymbol{\eta}^t$ для забезпечення $\tilde{x}^{t-1} + \boldsymbol{\eta}^t \in \mathcal{D}$ і $\|\boldsymbol{\eta}^t\|_2 = \delta \cdot d(x, \tilde{x}^{t-1})$, де $d(,)$ представляє функцію відстані між двома зразками, а δ - це гіперпараметр, який контролює масштаб збурення.
- Ортогональне збурення: проекція $\boldsymbol{\eta}^t$ на сферу навколо початкового введення x так, щоб $d(x, \tilde{x}^{t-1} + \boldsymbol{\eta}^t) = d(x, \tilde{x}^{-1})$.
- Рух у напрямку до початкового зображення так, щоб $\tilde{x}^{t-1} + \boldsymbol{\eta}^t \in \mathcal{D}$ і $d(x, \tilde{x}^{t-1}) - d(x, \tilde{x}^{-1} + \boldsymbol{\eta}^t) = \epsilon \cdot d(x, \tilde{x}^{t-1})$ обидва виконувались, де ϵ контролює розмір кроку руху.

Boundary Attack досягає продуктивності, порівнянної з найсучаснішими атаками білої коробки на DNN, навчених для класифікації. [39] представили Boundary Attack++, використовуючи бінарну інформацію на межі рішень для оцінки напрямку градієнта. Вони показали, що Boundary Attack++ вима-

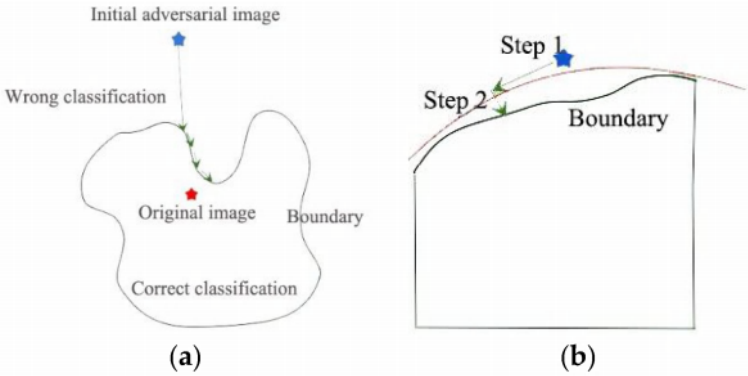


Рис. 1.2: Ілюстрація Boundary Attack. По суті, Boundary Attack виконує відбір зразків на межі між адверсаріальними та неадверсаріальними зображеннями [37].

гає значно менше запитів до моделі, ніж Boundary Attack. [41] запропонували атаку чорної скриньки на основі рішень, яка покращує ефективність запитів Boundary Attack, обмежуючи пошук адверсаріальних прикладів до низькочастотного домену. Існує багато інших методів, що належать до цієї категорії, такі як Guessing Smart [38].

Атаки на основі оптимізації: Замість застосування випадкових блукань, які не має жодних гарантій збіжності, дослідження виявили, що атаки на основі рішень також можуть бути сформульовані як розв'язання задачі оптимізації нульового порядку [44]. Вони показали, що як PGD, так і C&W функції втрат невизначені при атаках на основі рішень, і замість цього потрібно переформулювати проблему як пошук найкращого напрямку для адверсаріальних прикладів. Враховуючи x_0, функцію $g(\cdot)$, яка вимірює відстань між x_0 та межею рішення вздовж напрямку θ, можна визначити як:

$$g(\theta) = \arg\min_{\lambda>0} \big(f(x_0 + \lambda \frac{\theta}{\|\theta\|}) \neq y_0\big). \tag{1.1.18}$$

Проблему атаки можна тоді сформулювати як

$$\theta^* = \arg\min_{\theta} g(\theta), \tag{1.1.19}$$

і адверсаріальний приклад, який є найближчим до x_0, це $x^* = x_0 + g(\theta^*)\frac{\theta}{\|\theta\|}$. Хоча градієнт $g(\theta)$ не може бути безпосередньо обчислений, значення фун-

кції $g(\theta)$ можна обчислити за допомогою бінарного пошуку, тому стандартні розв'язувачі оптимізації нульового порядку можуть бути застосовані для розв'язання (1.1.19). В атаці OPT [30], метод Randomized Gradient-Free (RGF) [45] використовується для вирішення проблеми. Пізніше [42] показали, що знак градієнта з (1.1.19) можна обчислити більш ефективним способом, що призводить до покращеної атаки, званої Sign-OPT. З іншого боку, [40] запропонували ще одну оптимізаційну формулу, що призвело до ще одного ефективного алгоритму під назвою HotSkipJump Attack. Згодом, [46] запропонували RayS, яка реформулює задачу безперервної оптимізації в [30] у дискретну для атаки ℓ_∞ норми.

L-BFGS. Проблему знаходження адверсаріального прикладу можна сформулювати як задачу оптимізації з обмеженнями:

$$\min_{x':C(x')\neq C(x)} \|x - x'\|_2,$$

де x — це зображення, а C — класифікатор, що перетворює вектор значень пікселів зображення в дискретну множину міток [12]. Зазначимо, що x' може бути не єдиним, і його пошук є складною задачею.

Тому, Сегеді та ін. запропонували вирішувати іншу задачу [3], використовуючи алгоритм обмеженої пам'яті Бройдена-Флетчера-Голдфарба-Шанно (L-BFGS):

$$\min_{x'} \Big(c\|x - x'\|_2 + loss_{C,x}(x') \Big),$$

де $loss$ — це диференційована опукла функція втрат.

Атака одного пікселя. Атака одного пікселя є вкрай обмеженим сценарієм, коли зміна одного пікселя може змінити мітку цільового класу. Для цього було запропоновано алгоритм створення адверсаріальних прикладів, заснований на диференціальній еволюції [47].

Це завдання формалізовано як задачу оптимізації з обмеженнями:

$$\max_{e(x):\|e(x)\|_0\leqslant 1} f_{adv}(x + e(x)),$$

де $x = (x_1, .., x_n)$ є оригінальним зображенням, правильно класифікованим як клас t, $f_{adv}(x)$ є ймовірністю того, що x класифікується з міткою adv, а вектор $e(x) = (e_1, .., e_n)$ є адитивним адверсаріальним збуренням.

Кандидатні рішення (адверсаріальні зміни) кодуються в масив, який оптимізується методом диференціальної еволюції. Одне кандидатне рішення містить фіксовану кількість збурень, і кожне збурення є кортежем, що містить п'ять елементів: координати x-y і значення RGB збурення. Одне збурення змінює один піксель. Початкова кількість кандидатних рішень (популяція) становить 400, і на кожній ітерації створюється ще 400 кандидатних рішень (нащадків) за допомогою цієї формули:

$$x_i(g+1) = x_{r_1}(g) + F(x_{r_2}(g) - x_{r_3}(g))$$

де x_i є елементом кандидатного рішення, r_1, r_2, r_3 є випадковими числами, F є параметром масштабування, встановленим на рівні 0.5, g є поточним індексом покоління.

Адверсаріальний патч Було створено атаку, яка не намагається непомітно перетворити існуюче зображення на інше. Замість цього, ця атака створює патч, незалежний від зображення, який є надзвичайно помітним для нейронної мережі [48]. Цей патч можна розмістити під будь-яким кутом до класифікатора, і він змушує класифікатор видавати цільовий клас. Цей тип атаки є дуже потужним, оскільки не вимагає попередніх знань про освітлення, кут камери або модель класифікатора.

Алгоритм навчений максимізувати $\log(P[\hat{y}|\hat{x}])$ за умови, що $||x - \hat{x}||_\infty \leqslant \varepsilon$, де $P[y|x]$ є класифікатором, $x \in R^n$ — вхідне зображення, y — цільовий клас, $\hat{x}$, $\hat{y}$ — адверсаріальний вхід і бажаний вихід.

1.2. Огляд існуючих методів для виявлення адверсаріальних прикладів

Незважаючи на суперечки щодо того, чи є методи виявлення адверсаріальних прикладів захисними методами, вважається, що вони мають різні конкретні цілі, але узгоджуються з більшою метою протидії атакам. Захисні методи спрямовані на те, щоб класифікувати чисті зразки та їхні адверсаріальні версії з однаковим класом передбачення, тоді як методи виявлення спрямовані на класифікацію вхідних даних як адверсаріальних чи ні. Як зазначено в [16], жоден захист не зміг правильно класифікувати адверсаріальні приклади, і деякі дослідження спрямовані на розробку методів виявлення. Хоча методи виявлення вразливі до добре розроблених атак, вони можуть мати додаткову цінність навіть при використанні надійного захисного класифікатора. Наприклад, вихідні дані базового класифікатора можуть не збігатися з результатами надійного класифікатора, і потрібно знати, чи є це через адверсаріальний приклад чи ні.

Загалом, детектори відрізняються за двома факторами [49]: 1) використання знань про адверсаріальні атаки або їх відсутність, і 2) техніка, яка використовується для розрізнення чистих і адверсаріальних вхідних даних. Тому спочатку класифікуються методи детектування за першим фактором, а потім за другим. Для оцінки ефективності детектора використовують такі критерії:

Істинно-позитивний рівень (англ. True positive rate, TPR): Це точність детектора, яка вимірюється кількістю успішних ае, передбачених детектором, поділеною на загальну кількість успішних ае. Чим вище, тим краще.

Хибно-позитивний рівень (англ. False positive rate, FPR): Це дуже важливий критерій, який показує, наскільки детектор розглядає чисті вхідні дані як адверсаріальні. Вимірюється кількістю чистих вхідних даних, виявлених як адверсаріальні, поділеною на загальну кількість чистих вхідних даних. Чим нижче, тим краще.

Складність: Це час, необхідний для тренування детектора. Деякі галузі мають достатньо апаратних можливостей для запуску детекторів з високою обчислювальною складністю, але в разі наявності нових даних або необхідності включення нових атак недоцільно тренувати дуже великі моделі багато разів.

Навантаження: Це стосується архітектури детектора та розміру додаткових параметрів, необхідних для розгортання детектора. Чим менше, тим краще, щоб бути придатним для платформ з обмеженими ресурсами пам'яті та обчислень, таких як мобільні пристрої.

Час інференсу: Це час роботи детектора для визначення, чи є вхідні дані адверсаріальними. Для реальних застосувань важливо, щоб інференс працював швидко.

1.2.1. Виявлення на основі supervised підходу

У supervised виявленні захисник враховує адверсаріальні приклади, згенеровані одним або кількома алгоритмами адверсаріальних атак, при розробці та навчанні детектора. Вважається, що адверсаріальні приклади мають відмінні характеристики, які відрізняють їх від чистих вхідних даних [50], тому захисники використовують це для створення надійного детектора. Для досягнення цього в літературі було представлено багато підходів.

1.2.1.1. Використання допоміжної моделі

У цьому підході моделі використовують характеристики, які можна витягнути, контролюючи поведінку чистих та адверсаріальних зразків. Потім будуються і обчислюються або класифікатори, або пороги.

Невизначеність моделі. Захисники використовують невизначеність моделей глибокого навчання для чистих та адверсаріальних вхідних даних. Невизначеність зазвичай вимірюється додаванням випадковості до моделі за допомогою техніки Dropout [51]. Ідея полягає в тому, що при високому dropout

передбачення класу для чистих даних залишається правильним, тоді як для адверсаріальних прикладів це не так. Значення невизначеності використовуються як характеристики для побудови бінарного класифікатора як детектора. Feinman та ін. [52] запропонували метрику баєсівської невизначеності (англ. Bayesian Uncertainty, BU), яка використовує монте-карло dropout для оцінки невизначеності, для виявлення тих адверсаріальних прикладів, які знаходяться поблизу маніфолду класів, тоді як Smith та ін. [53] використали метод взаємної інформації (англ. mutual information) для такої задачі.

На основі softmax/логітів. Hendrycks та ін. [54] показали, що ймовірносний розподіл після softmax шару можуть бути використані для виявлення аномалій. Вони додали декодер для відновлення чистого вхідного сигналу з softmax і тренували його спільно з базовим класифікатором. Потім вони навчили класифікатор, детектор , використовуючи відновлені вхідні дані, логіти і значення впевненості для чистих і адверсаріальних вхідних даних. У одному з методів, запропонованих у [55], Pertigkiozoglou та ін. використовували векторні характеристики моделі, тобто значення впевненості, для обчислення регуляризованих векторних характеристик. Базовий класифікатор був перенавчений шляхом додавання цих регуляризованих векторних характеристик до останнього шару класифікатора. Детектор вважає вхідний сигнал адверсаріальним, якщо немає збігу між базовим класифікатором і перенавченим класифікатором. Aigrain та ін. [56] побудували простий нейронний детектор, який використовує логіти базової моделі для чистих і адверсаріальних прикладів як вхідні дані для побудови бінарного класифікатора. Нарешті, виходячи з гіпотези, що різні моделі роблять різні помилки при поданні одних і тих самих атакувальних вхідних даних, Monteiro та ін. [57] запропонували метод виявлення невідповідності бі-моделі. Детектор є бінарним RBF-SVM класифікатором, вхідними даними якого є результати двох базових класифікаторів для чистих і адверсаріальних прикладів.

Класифікатор на основі сирих адверсаріальних прикладів. Gong та

ін. [58] навчили бінарний класифікатор, детектор, який повністю відділений від базового класифікатора і приймає як чисті вхідні дані так і адверсаріальні зображення. У [59, 60] автори перенавчили базовий класифікатор з новим доданим класом, тобто адверсаріальним класом. Hosseini та ін. використали адверсаріальне навчання, а мітки для навчання були зроблені за допомогою згладжування міток (англ. label smoothing) [61]. В одному з методів, запропонованих у [55], автори скористалися частинами вхідних даних моделі глибокого навчання, які ігноруються моделлю для виявлення адверсаріальних прикладів. Вони ітеративно змінювали вхідні дані, чисті або адверсаріальні, і якщо ймовірність передбачуваного класу для вхідного сигналу була менша за поріг, то вхідний сигнал вважався адверсаріальним.

На основі NSS. Метод статистик натуральних сцен (англ. Natural Scene Statistics, NSS) використовувався у багатьох областях обробки зображень, особливо в оцінці якості зображень, оскільки було доведено, що статистика природних зображень відрізняється від статистики маніпульованих зображень. Kherchouche та ін. [62] дотримувалися цього припущення і побудували бінарний класифікатор, який приймає як вхідні дані параметри розподілу узагальненої гауссової функції (GGD) та асиметричного узагальненої функції (AGGD), обчислені з коефіцієнтів MSCN [63] для чистих зображень та адверсаріальних прикладів на основі PGD.

На основі градієнта. Lust та ін. [64] запропонували детектор під назвою GraN. На кожному шарі вони обчислювали норму градієнта згладженого вхідного сигналу, чистого та адверсаріального, щодо передбачуваного класу базового класифікатора. Потім вони тренували бінарний класифікатор для виявлення адверсаріальних прикладів під час інференції.

Erase&restore (E&R) [65]. У цій моделі Zuo та ін. запропонували бінарний класифікатор, детектор, для тренування чистих та адверсаріальних зразків L_2-норми після обробки. Спочатку вхідні зразки оброблялися шляхом видалення деяких пікселів та їх відновлення в процесі інтерполяції. По-друге,

ймовірність впевненості обчислювалася за допомогою базового класифікатора. Нарешті, оброблена ймовірність впевненості передавалася бінарному класифікатору. Детектор оголошував вхідний сигнал адверсаріальним, якщо бінарний класифікатор вказував на це.

1.2.1.2. Статистичний підхід

У цьому підході обчислюються різні статистичні властивості чистих і адверсаріальних вхідних даних, які використовуються для побудови детектора. Ці властивості більшою мірою пов'язані з розподілом/маніфолдами навчальних даних або поза ними. У літературі використовуються наступні статистичні підходи:

Maximum mean discrepanc (MMD). Grosse та ін. [59] застосували статистичний тест, під назвою mmd [66], для розрізнення адверсаріальних прикладів від навчальних даних моделі. Це модельно-незалежний і ядровий тест на дві вибірки. Щоб відповісти на припущення гіпотезного тесту, детектор спочатку обчислює mmd між чистими та адверсаріальними зразками, $a = MMD(x, x')$, де MMD визначена наступним чином:

$$\mathrm{MMD}_b[\mathcal{F}, X_1, X_2] = \sup_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^{n} f(x_{1i}) - \frac{1}{m} \sum_{i=1}^{m} f(x_{2i}) \right)$$

де $x_{1i} \in X_1$ є i-м елементом першої вибірки. $x_{2j} \in X_2$ є j-м елементом другої вибірки, яка можливо отримана з іншого розподілу, ніж X_1. $f \in \mathcal{F}$ є ядерною функцією, обраною для максимізації відстаней між вибірками з двох розподілів. У нашому випадку використовується Гауссівське ядро.

Потім елементи x та x' перемішуються у два нові набори y_1 та y_2, і обчислюється $b = MMD(y_1, y_2)$. Нарешті, робиться висновок, що x та x' походять з різних розподілів, і гіпотеза відхиляється, якщо $a < b$.

Метод головних компонент (англ. Principal component analysis, PCA). У роботі [67] було побудовано каскадні класифікатори. Кожен класифікатор svm відповідає одному шару. Він тренується на чистих і адверсаріальних

зразках. Вхід svm - це pca виходу кожного шару. Детектор оголошує вхідний сигнал чистим, якщо всі класифікатори так вважають.

Kernel Density (KD). Було показано, що підпростори адверсаріальних прикладів зазвичай мають нижчу щільність, ніж чисті зразки, особливо якщо вхідний зразок знаходиться далеко від маніфолду класу. Feinman та ін. [52] запропонували оцінку KD для кожного класу в навчальних даних і потім навчили бінарний класифікатор, детектор , використовуючи характеристики щільності та невизначеності чистих, шумових і адверсаріальних зразків.

Local intrinsic dimensionality (LID). Як альтернативний підхід до KD, [68] запропонували використати LID для обчислення розподілу відстаней вхідного зразка до його сусідів для оцінки здатності заповнення простору навколо цього вхідного зразка.

У теорії внутрішньої розмірності класичні моделі розширення (такі як розмірність розширення та узагальнена розмірність розширення [69, 70]) вимірюють швидкість зростання кількості об'єктів даних у міру збільшення відстані від контрольної вибірки. Наприклад, у евклідовому просторі об'єм m-вимірної кулі зростає пропорційно до r^m при масштабуванні її розміру на фактор r. Виходячи з цієї швидкості зростання об'єму з відстанню, розмірність розширення m можна визначити як:

$$\frac{V_2}{V_1} = \left(\frac{r_2}{r_1}\right)^m \Rightarrow m = \frac{\ln(V_2/V_1)}{\ln(r_2/r_1)}. \tag{1.2.1}$$

Використовуючи ймовірнісну масу як проксі для об'єму, класичні моделі розширення забезпечують *локальний огляд* структурної розмірності даних, оскільки їх оцінка обмежується околицею навколо вибірки інтересу. Перенесення концепції розмірності розширення на статистичні умови безперервних розподілів відстаней призводить до формального визначення LID [71].

Нехай $x \in X$ є вибіркою даних, і $R > 0$ є випадковою величиною, що позначає відстань від x до інших вибірок даних. Якщо функція розподілу $F(r)$

випадкової величини R є позитивною і безперервно диференційованою на відстані $r > 0$, тоді локальна внутрішня розмірність x на відстані r визначається як:

$$\mathrm{LID}_F(r) \triangleq \lim_{\epsilon \to 0} \frac{\ln\big(F((1+\epsilon)\cdot r)/F(r)\big)}{\ln(1+\epsilon)} = \frac{r \cdot F'(r)}{F(r)}, \tag{1.2.2}$$

якщо цей ліміт існує.

$F(r)$ аналогічна до об'єму V у рівнянні (1.2.1); однак, зауважимо, що основна міра відстані не обов'язково повинна бути евклідовою. Остання рівність у рівнянні (1.2.2) випливає з застосування правила Л'Опіталя до лімітів [71]. Локальна внутрішня розмірність у точці x визначається як ліміт, коли радіус r прагне до нуля:

$$\mathrm{LID}_F = \lim_{r \to 0} \mathrm{LID}_F(r). \tag{1.2.3}$$

LID_F описує відносну швидкість збільшення кумулятивної функції відстані $F(r)$ у міру збільшення відстані r від нуля, і може бути оцінена за допомогою відстаней від x до його k найближчих сусідів у вибірці [72].

На основі відстані Махаланобіса. Як альтернативний захід до kd та lid, Lee та ін. [73] запропонували оцінку на основі відстані Махаланобіса для виявлення зразків, що не належать до розподілу, та адверсаріальних вхідних зразків. Цей показник впевненості базується на індукованому генеративному класифікаторі за gda, який фактично замінює класифікатор softmax.

knn. У роботі [74] спочатку вимірювали вплив кожного навчального зразка на дані валідаційного набору, а потім знаходили найбільш підтримуючі навчальні зразки для будь-якого заданого валідаційного прикладу. Потім, на кожному шарі, використовуючи репрезентативний вихід шарів глибокого навчання, модель knn підганяється для ранжування цих підтримуючих навчальних зразків. Ці характеристики витягуються з чистих і адверсаріальних прикладів для тренування детектора. Згодом у роботі [75] було запропонова-

но детектор Neighbor Context Encoder (NCE). Він використовував трансформер [76] для навчання класифікатора з k найближчими сусідами для представлення оточуючого підпростору виявленого зразка.

1.2.1.3. Інваріантний підхід до моделі

Вважається, що чисті та адверсаріальні зразки генерують різні карти ознак і різні значення активації для шарів мережі. Аналіз порушення цього мережевого інваріанту є основним компонентом багатьох методів виявлення.

Safetynet [77]. SafetyNet висуває гіпотезу «Адверсаріальні атаки працюють, створюючи інші шаблони активації на пізніх етапах ReLU порівняно з тими, що створюються природними прикладами». Тому SafetyNet квантує останній шар активації ReLU моделі та будує бінарний svm rbf класифікатор.

Динамічне тренування проти ворога [78]. Metzen та ін. представили динамічне тренування проти ворога для зміцнення детектора, в якому класифікатор тренувався з адверсаріальними прикладами. Детектор доповнюється переднавченим класифікатором на виході певного шару. Він використовує представницький вихід шару для чистих зразків та для адверсаріальних прикладів, які генеруються на льоту, щоб побудувати бінарний класифікатор.

На основі гістограм [55]. Pertigkiozoglou та ін. виявили, що для адверсаріальних прикладів збільшуються значення деяких піків чистого виходу, тоді як значення інших точок виходу зменшуються. Тому вони побудували бінарний svm класифікатор, який використовує гістограму виходу першого шару згортки базового класифікатора для чистих та адверсаріальних прикладів.

Еволюція адверсаріальних прикладів [79]. Carrara та ін. висунули гіпотезу, що проміжні представлення адверсаріальних прикладів мають іншу еволюцію порівняно з чистими вхідними даними. Детектор кодує відносні положення внутрішніх активацій точок, що представляють щільні частини простору ознак. Детектор є бінарним класифікатором, побудованим на основі попередньо натренованої мережі, і використовує кодування відносних по-

ложень внутрішніх активацій точок для адверсаріальних і чистих вхідних даних.

RAID [80]. Eniser та ін. побудували бінарний класифікатор, який використовує різниці у значеннях активацій нейронів між чистими та адверсаріальними вхідними даними. Щоб ускладнити адаптивні атаки, автори також запропонували розширення до RAID під назвою Pooled-RAID. Це розширення має на меті тренувати пул детекторів, кожен з яких тренується з випадково вибраною кількістю нейронів. Під час тестування Pooled-RAID випадково вибирає один класифікатор з пулу для перевірки, чи є вхідні дані адверсаріальними чи ні.

1.2.2. Виявлення на основі unsupervised підходу

Основним обмеженням supervised методів є те, що вони вимагають попередніх знань про атаки і, отже, можуть бути неефективними проти нових/невідомих атак. У ненаглядовому виявленні захисник враховує лише чисті навчальні дані при проектуванні та тренуванні детектора. Це також відоме як моделі прогнозування непослідовності, оскільки вони залежать від того, що ае можуть не обдурити кожну модель nn. Ненаглядові детектори спрямовані на зменшення обмеженого простору ознак вхідних даних, доступного для противників, і для досягнення цієї мети було представлено багато підходів у літературі.

1.2.2.1. Підхід допоміжної моделі

На відміну від допоміжних моделей наглядового виявлення, ненаглядові моделі використовують ознаки, які можна отримати, спостерігаючи тільки поведінку чистих зразків. Потім будуються та обчислюються або класифікатори, або порогові значення.

Класифікатор knn [81]. Carrara та ін. використовували вихід одного з проміжних шарів моделі dl для побудови класифікатора knn. Вихід цього кла-

сифікатора не використовується для виявлення, але він використовується для оцінки передбаченого класу базового класифікатора. Детектор оголошує вхід адверсаріальним, якщо ця оцінка менша за певний поріг. Вони також надали процес використання pca виходу одного з проміжних шарів моделі dl для зменшення розміру ознаки.

Зворотна крос-ентропія [82]. Pang та ін. запропонували процедуру навчання та детектор на основі порогового значення.

Була розроблена функція втрат для підвищення стійкості класифікаторів DNN. Ключовим є примус класифікатора DNN відображати всі нормальні приклади в околиці низьковимірних многовидів $S_{\hat{y}}$ в прихованому просторі останнього шару. Цього можна досягти, зробивши немаксимальні елементи $F(x)$ максимально рівними, що забезпечує високе значення non-ME для кожного нормального входу. Для тренувальних даних (x, y), нехай R_y позначає зворотний вектор міток, де y-й елемент дорівнює нулю, а інші елементи рівні $\frac{1}{L-1}$. Один з очевидних способів стимулювання рівномірності серед немаксимальних елементів $F(x)$ є застосування методу регуляризації моделі, який називається згладжуванням міток szegedy2016rethinking, що можна здійснити, вводячи крос-ентропійний термін між R_y і $F(x)$ у CE цілі:

$$\mathcal{L}_{CE}^{\lambda}(x, y) = \mathcal{L}_{CE}(x, y) - \lambda \cdot R_y^{\top} \log F(x), \tag{1.2.4}$$

де λ є параметром компромісу. Проте, легко показати, що мінімізація $\mathcal{L}_{CE}^{\lambda}$ рівнозначна мінімізації крос-ентропії між $F(x)$ та L-вимірним вектором P^{λ}:

$$P_i^{\lambda} = \begin{cases} \frac{1}{\lambda+1}, & i = y, \\ \frac{\lambda}{(L-1)(\lambda+1)}, & i \neq y. \end{cases} \tag{1.2.5}$$

Зверніть увагу, що $1_y = P^0$ і $R_y = P^{\infty}$. Коли $\lambda > 0$, нехай $\theta_{\lambda}^* = {}_{\theta}\mathcal{L}_{CE}^{\lambda}$, тоді прогноз $F(x, \theta_{\lambda}^*)$ буде схильний дорівнювати P^{λ}, а не істинному значенню 1_y. Це спричиняє упередженість прогнозів. Для отримання неупереджених прогнозів, які схиляють вектор виходу $F(x)$ до 1_y і одночасно стимулюють рівно-

мірність ймовірностей для неправдивих класів, ми визначаємо іншу цільову функцію на основі того, що ми називаємо *зворотна крос-ентропія (RCE)* як

$$\mathcal{L}_{CE}^{R}(x, y) = -R_y^\top \log F(x). \tag{1.2.6}$$

Мінімізація RCE еквівалентна мінімізації $\mathcal{L}_{CE}^{\infty}$. Зауважте, що безпосередньо мінімізуючи $\mathcal{L}_{CE}^{R}$, тобто $\theta_R^* = {}_\theta \mathcal{L}_{CE}^{R}$, ми отримаємо зворотний класифікатор $F(X, \theta_R^*)$, що означає, що при поданні вхідного x, зворотний класифікатор $F(X, \theta_R^*)$ буде не тільки схильний призначати найнижчу ймовірність справжньому класу, але й схильний видавати рівномірний розподіл на інші класи. Це просте розуміння призводить до нашої повної процедури навчання RCE, яка складається з двох частин, як описано нижче:

Зворотне тренування: Даний тренувальний набір $\mathcal{D} := \{(x^i, y^i)\}_{i \in [N]}$, навчання DNN $F(X, \theta)$ бути зворотним класифікатором шляхом мінімізації середньої RCE втрати: $\theta_R^* = {}_\theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{CE}^{R}(x^i, y^i)$.

Зворотні логіти: Знак $-$ для кінцевих логітів, які подаються в softmax шар як $F_R(X, \theta_R^*) = (-Z_{pre}(X, \theta_R^*))$.

Тоді отримаємо мережу $F_R(X, \theta_R^*)$, яка повертає звичайні прогнози на класи, і $F_R(X, \theta_R^*)$ називається мережею, навченою через *процедуру навчання RCE*.

На основі невизначеності. Відповідно до припущення bu, що відстані ae від даних у розподілі роблять невизначеність моделі dl відмінною від чистих даних, Sheikholeslami та ін. [**?**] запропонували додати випадковість для випадково вибраних прихованих одиниць кожного шару моделі dl. Потім оцінюється невизначеність для даних навчання у розподілі, і визначається порогове значення на основі взаємної інформації. Вони надали розв'язувач з мінімальною варіацією для оцінки невизначеності на рівні шару. Під час інференції загальна невизначеність вхідного зображення оцінюється за допомогою виходів прихованих шарів. Детектор оголошує вхідний зразок адверсаріальним, якщо його взаємна інформація перевищує порогове значення.

Deep neural rejectio (DNR) [83]. Sotgiu та ін. запропонували використовувати виходи останніх N репрезентативних шарів базового класифікатора для побудови N класифікаторів svm з rbf ядром. Виходи цих класифікаторів, тобто ймовірності впевненості, комбінуються для побудови останнього класифікаційного класифікатора, який є класифікатором svm-rbf. Детектор оголошує адверсаріальні вхідні дані адверсаріальними, якщо максимальна ймовірність впевненості менше за попередньо визначений поріг.

Вибіркове виявлення [84]. Aldahdooh та ін. запропонували техніку sfad. Вони використовують сучасний метод невизначеності під назвою SelectiveNet [85] і інтегрують три модулі виявлення. Перший - це вибірковий модуль виявлення, який є виявленням на основі порогового значення, отриманим з невизначеності чистих навчальних даних за допомогою SelectiveNet. Другий - модуль виявлення впевненості, який є виявленням на основі порогового значення, отриманим з ймовірностей softmax чистих навчальних даних з класифікаторів sfad. Класифікатори sfad аналізують представницькі дані останніх N шарів як ключову точку для подання надійних ознак вхідних даних, використовуючи автоенкодери, зразкове підвищення/пониження, вузьке місце та блоки шуму. Останній модуль - це ансамбльне передбачення, яке є прогнозуванням невідповідності між детектором та базовими класифікаторами dl.

1.2.2.2. Статистичний підхід

У цьому підході розраховуються різні статистичні властивості лише чистих вхідних даних, які потім використовуються для побудови детектора. Ці властивості більше пов'язані з розподілом даних тренувального набору або виходом за його межі. У літературі використовуються наступні статистичні підходи:

Розподіл Softmax [54]. Hendrycks та ін. виявили, що ймовірність максимальної/передбачуваної класи чистих зразків у розподілі вища, ніж у зразків

поза розподілом. Ця інформація використовується для розрахунку дивергенції Кульбака-Лейблера [**?**] між чистими зразками та зразками у розподілі для визначення порогу.

pca [86]. Hendrycks та ін. спостерігали, що дисперсія пізніших компонент pca ae більша, ніж у чистих вхідних даних, тому вони запропонували детектор, який оголошує вхід адверсаріальним, якщо дисперсія пізніших компонент pca перевищує поріг.

Gaussian Mixture Models (GMM) [87]. Zheng та ін. запропонували метод виявлення під назвою I-defender, який досліджує розподіли прихованих станів моделі dl на основі чистих тренувальних даних. I-defender використовує gmm для наближення внутрішнього розподілу прихованих станів кожного класу. I-defender моделює стан лише повністю зв'язаних прихованих шарів і розраховує поріг для кожного класу. Детектор оголошує вхідний зразок адверсаріальним, якщо його розподіл прихованого стану менше порогу передбаченого класу. У роботі [88] робиться припущення, що адверсаріальні зразки мають низьку ймовірність щодо моделі густини передбаченого класу (”надто нетипові”) або високу ймовірність для класу, відмінного від класу чистого зразка (”надто типові”). Остаточний бал для ”надто нетипових”та ”надто типових”обчислюється за допомогою дивергенції Кульбака-Лейблера. Детектор оголошує вхідний зразок адверсаріальним, якщо бал перевищує визначений поріг.

1.2.2.3. Підхід з використанням денойзера

Щоб запобігти точної оцінки місця розташування ae, можна зробити градієнт вхідного сигналу дуже малим або неправильно великим. Це явище відоме як вибухові/затухаючі градієнти. Один із способів цього досягти - це денойз або реконструкція ae для максимізації здатності проектувати ae на маніфолд тренувальних даних. Основним обмеженням використання денойзера є те, що він не гарантує видалення всього шуму і може вносити додаткові спо-

творення. Крім того, він не ефективний для денойзу атак L_0, оскільки атаки L_0 націлені на кілька пікселів, які можуть не бути денойзовані денойзером.

PixelDefend [89]. Генеративні моделі, такі як PixelCNN [90], вибухають градієнт шляхом застосування кумулятивного добутку часткових похідних кожного шару. PixelDefend виявлення [89] використовує PixelCNN для побудови детектора. Спочатку PixelDefend реконструює/очищує чисті тренувальні дані за допомогою PixelCNN і потім обчислює ймовірності передбачення за допомогою базового класифікатора. Виявлено, що реконструйовані зображення мають вищі ймовірності у розподілі тренувальних даних. Потім обчислюється щільність ймовірності тренувальних зразків. Детектор працює шляхом обчислення щільності ймовірності перевіреного вхідного сигналу. Потім ця щільність порівнюється з щільностями тренувальних даних. Нарешті, ранг використовується як тестова статистика, і обчислюється p-значення для визначення, чи належить вхідний зразок до розподілу тренувальних даних або є адверсаріальним.

Magnet [91]. Magnet навчає денойзери на чистих тренувальних даних для реконструкції вхідних зразків. Magnet запропонував два способи виявлення ae. Перший спосіб припускає, що помилка реконструкції буде малою для чистих зображень і великою для ae, тому він обчислює помилку реконструкції як бал. Другий спосіб вимірює відстані між прогнозами вхідних зразків та їх денойзованими/фільтрованими версіями. Детектор оголошує вхідний зразок адверсаріальним, якщо бал перевищує визначений поріг.

1.2.2.4. Підхід з усуненням ознак

Цей підхід спрямований на видалення непотрібних ознак вхідних зразків для знищення збурень. Цей процес обмежить простір ознак, доступний для супротивника, але якщо стиснювач не побудовано ефективно, він може збільшити збурення.

Зменшення глибини бітів та згладжування [92]. Xu та ін. стискають

вхідні зразки шляхом проектування/перетворення їх для отримання нових зразків. Вони використовували зменшення глибини кольору, локальне згладжування за допомогою медіанного фільтра та нелокальне згладжування за допомогою нелокального середнього денойзера. Детектор оголошує вхідний сигнал адверсаріальним, якщо відстань між передбаченим оригінальним входом та стислою версією перевищує визначений поріг.

Адаптивне зменшення шуму [93]. Liang та ін. стискають вхідні зразки за допомогою скалярної квантизації та просторового фільтра згладжування. Вони використовували ентропію зображення як метричу для реалізації адаптивного зменшення шуму. Детектор оголошує вхідний сигнал адверсаріальним, якщо клас оригінального входу відрізняється від стислої версії.

1.2.2.5. Підхід на основі мережевих інваріантів

На відміну від підходу на основі мережевих інваріантів при наглядовому виявленні, тут детектор спрямований на спостереження за поведінкою чистих тренувальних даних лише у проміжних шарах моделі dl. Робота [94] показала, що якщо моніторити два канали атаки, канал походження та канал розподілу значень активації, то можна виявити ae. Ma та ін. [94] запропонували метод nic, який будує набір моделей для окремих шарів, щоб описати канали походження та розподілу значень активації. Канал походження описує нестабільність набору активованих нейронів у наступному шарі при наявності малих змін у вхідному зразку, тоді як канал розподілу значень активації описує зміни зі значеннями активації шару. Для навчання інваріантних моделей автори використовували задачу класифікації одного класу (OCC) як спосіб моделювання тренувальних даних у розподілі. Детектор є спільним OCC класифікатором, що об’єднує всі виходи інваріантних моделей. Він оголошує вхідний зразок адверсаріальним, якщо класифікатор детектора заявляє, що вхід виходить за межі розподілу.

1.2.2.6. Об'єктно-орієнтований підхід

У цьому підході метою є витяг об'єктно-орієнтованих ознак з вхідного зразка та порівняння їх з тренувальними даними з тим самим прогнозованим ярликом. UnMask — це метод, запропонований Freitas та ін. [95], який працює наступним чином: по-перше, припустимо, що супротивник змінив зображення велосипеда, щоб його передбачили як птаха. UnMask спочатку витягує об'єктно-орієнтовані низькорівневі ознаки із зображення атаки «велосипед» і порівнює їх з об'єктно-орієнтованими низькорівневими ознаками «птаха». Якщо перекриття невелике, детектор оголосить вхід адверсаріальним. Крім того, Unmask продовжує «як захист» знаходити, який клас у тренувальних даних має найбільше перекриття з передбаченим, щоб оголосити правильний клас.

1.2.3. Баєсові нейронні мережі

Ідея Баєсових нейронних мереж (англ. Bayesian Neural Networks, BNN) ілюструється на Рисунку 1.3. У [96] автор представив ефективний алгоритм для навчання параметрів BNN. Враховуючи спостережувані випадкові змінні (x, y), BNN прагне оцінити розподіли прихованих змінних w замість оцінювання максимальної ймовірності w_{MLE} для ваг. З точки зору Байєса, кожен параметр тепер є випадковою змінною, що вимірює невизначеність оцінки, модель потенційно може витягнути більше інформації для підтримки кращого прогнозування (з точки зору точності, стійкості тощо).

Враховуючи вхід x і клас y, BNN прагне оцінити апостеріорний розподіл ваг $p(w|x, y)$, враховуючи апостеріорний $p(w)$. Справжній апостеріорний розподіл можна наблизити параметричним розподілом $q_{\boldsymbol{\theta}}(w)$, де невідомий параметр $\boldsymbol{\theta}$ оцінюється шляхом мінімізації розбіжності Кульбака-Лейблера (KL розбіжності):

$$\mathsf{KL}\big(q_{\boldsymbol{\theta}}(w) \parallel p(w|x, y)\big) \tag{1.2.7}$$

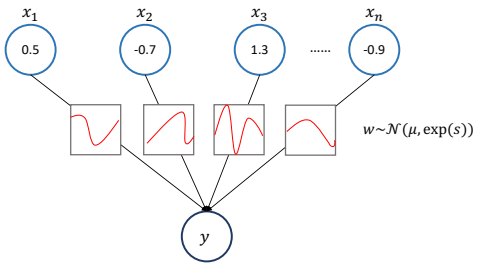


Рис. 1.3: Ілюстрація Баєсової нейронної мережі [97]. Усі ваги в BNN представлені ймовірнісними розподілами можливих значень і не мають єдиного фіксованого значення. Червоні криві на графіку представляють розподіли. BNN розглядається як ймовірнісна модель: заданий вхід, BNN призначає ймовірність кожному можливому виходу y, використовуючи набір параметрів, вибраних із навчених розподілів.

по відношенню до $\boldsymbol{\theta}$. Для спрощення часто припускають, що $q_{\boldsymbol{\theta}}$ є повністю факторизованим нормальним розподілом:

$$q_{\boldsymbol{\theta}}(w) = \prod_{i=1}^{d} q_{\boldsymbol{\theta}_i}(w_i), \text{ і } q_{\boldsymbol{\theta}_i}(w_i) = \mathcal{N}(w_i; \boldsymbol{\mu}_i, \exp(s_i)^2), \tag{1.2.8}$$

де μ та s є параметрами нормальних розподілів ваг. Цільова функція для навчання BNN переформульована з виразу (1.2.7) і показана у виразі (1.2.10), що є сумою частини, залежної від даних, і частини регуляризації:

$$\arg\max_{\mu,s} \Big\{ \sum_{(x_i,y_i)\in \boldsymbol{D}} E_{w\sim q_{\mu,s}} \log p(y_i|x_i, w) \tag{1.2.9}$$

$$- \mathsf{KL}\big(q_{\mu,s}(w) \parallel p(w)\big) \Big\}, \tag{1.2.10}$$

де $\boldsymbol{D}$ представляє розподіл даних. У першому терміні цільової функції (1.2.10), ймовірність y_i, враховуючи x_i і ваги, є виходом моделі. Ця частина представляє втрати при класифікації. Другий термін цільової функції (1.2.10) намагається мінімізувати розбіжність між апріорним та параметричним розподілом, що можна розглядати як регуляризацію [96]. Автор [98] показав, що апостеріорний середній градієнт BNN робить його більш стійким

до градієнтних атак, ніж DNN. Хоча ідея використання BNN для підвищення стійкості до адверсаріальних прикладів не нова [99,100], попередні роботи не використовували BNN для виявлення адверсаріальних прикладів. У [99, 100] BNN було поєднано з адверсаріальним тренуванням [11] для підвищення точності стійкої класифікації.

1.3. Запропонований метод для адверсаріального виявлення

Запропонований алгоритм базується на припущеннях попередніх досліджень про геометричні властивості адверсаріальних прикладів [7] і використовує ідею, що кожен алгоритм атаки оптимізує відстань, використовуючи лише певні метрики відстані. При наявності достатньої кількості різних норм відстаней як ознак, припускається, що адверсаріальні введення будуть відрізнятися від реальних.

Для представлення даних у низьковимірному просторі було використано навчану згорткову нейронну мережу-автоенкодер. Маючи таке низьковимірне представлення, ми можемо використовувати ідею, що дані концентруються навколо низьковимірного многовиду. Ми припускаємо, що реальні дані належать гладкому многовиду, а адверсаріальні введення — ні.

Маючи низьковимірне представлення, алгоритм витягує ознаки з даних. Обчислені ознаки — це відстані L_1, L_2 і L_{inf} до відповідних центроїдів у низьковимірному просторі. Останнє припущення полягає в тому, що для кожної точки поза многовидом відстань до центроїда найближчих сусідів (KNN) більша, ніж відстань до центроїда для даних на многовиді. Останній крок алгоритму для виявлення адверсаріального введення — навчання SVM-бінарного класифікатора з ознаками відстаней.

1.4. Результати

Для тестування даного алгоритму було вирішено використовувати набори даних MNIST і CIFAR-10.

У таблиці нижче представлені архітектури згорткових автоенкодерів (частина енкодера) для обох наборів даних, метою яких було виконання нелінійного зменшення розмірності:

MNIST		CIFAR-10	
Назва шару	Розмір виходу	Назва шару	Розмір виходу
Conv1	28x28x4	Conv1	16x16x12
Batch norm	28x28x4	Batch norm	16x16x12
Relu	28x28x4	Relu	16x16x12
Max pooling	14x14x4	Conv2	8x8x24
Conv1	14x14x8	Batch norm	8x8x24
Batch norm	14x14x8	Relu	8x8x24
Relu	14x14x8	Conv3	4x4x36
Max pooling	7x7x8	Batch norm	4x4x36
Conv1	7x7x16	Relu	4x4x36
Batch norm	7x7x16	Conv4	2x2x48
Relu	7x7x16	Batch norm	2x2x48
Max pooling	4x4x16	Relu	2x2x48
Conv1	1x1x32	Середнє пулінгування	1x1x48
Batch norm	1x1x32	-	-
Relu	1x1x32	-	-

Таблиця 1.1

Архітектури згорткових автоенкодерів

У таблиці 1.2 показані обчислені середні відстані від зразків до центроїдів:

Набір даних	Алгоритм	L_1	L_2	L_{inf}
MNIST	Реальні	0.29	0.29	0.31
	FGS	0.40	0.39	0.43
	DeepFool	0.35	0.34	0.38
	OnePixel	0.38	0.37	0.42
CIFAR-10	Реальні	0.02961	0.028244	0.03206
	FGS	0.02972	0.028284	0.0323
	DeepFool	0.02973	0.02821	0.03213
	OnePixel	0.02971	0.02819	0.032187

Таблиця 1.2

Відстані від зразків до центроїдів для різних метрик відстані

Обчислені середні значення показують чітке розділення в наборі даних MNIST і порівняно мале розділення для CIFAR-10.

Для навчання класифікатора SVM були складені нові набори даних з реальними та адверсаріальними зображеннями. Складені результати показані в таблиці 1.3.

Набір даних	Реальні	FGS	DeepFool	One-Pixel
MNIST	5000	2418	2417	165
CIFAR-10	5000	1667	1667	1666

Таблиця 1.3

Адверсаріальні набори даних для MNIST і CIFAR-10

Набір даних	Точність	F1	Точність (Precision)	Повторюваність (Recall)
MNIST	0.65	0.65	0.65	0.65
CIFAR-10	0.49	0.53	0.49	0.58

Таблиця 1.4

Результати класифікації

Результати класифікатора представлені в таблиці 1.4:
Виявлено, що для набору даних MNIST класифікатор SVM зміг визначити деяке розділення між адверсаріальними та оригінальними даними, хоча для CIFAR-10 він не спрацював.

1.5. Висновки до розділу

Після проведення експерименту описаного вище, початкова гіпотеза про розділення між адверсаріальними та оригінальними даними була частково проілюстрована в розділі результатів, хоча результати класифікації були недостатньо хорошими через високу варіативність в розподілі відстаней від зразків до центроїдів.
Було виявлено, що адверсаріальний розділ залежить від властивостей низьковимірного простору та архітектури нейронної мережі, яка проектує вхідні зображення. Наприклад, додавання шару нормалізації покращило точність класифікації на 5% для набору даних MNIST.
Тим не менш, поточний алгоритм може бути вдосконалений шляхом додавання додаткових геометричних властивостей та вибору більш відповідної архітектури мережі для зменшення розмірності.

РОЗДІЛ 2
МОДЕЛЮВАННЯ СИГНАЛІВ З ДОПОМОГОЮ МЕХАНІЗМУ УВАГИ З РУХОМИМ СЕРЕДНІМ

У цьому розділі оцінюється продуктивність блоку нейронної мережі - MEGA (Moving Average Equipped Gated Attention) [2] у завданні моделювання аудіо мовлення. Метою цих експериментів є всебічна оцінка параметризації MEGA для моделей послідовностей. Блок MEGA був успішно застосований у задачах комп'ютерного зору, і тепер ми хочемо перевірити його ефективність у задачах, пов'язаних із обробкою мовлення. Ці експерименти допоможуть визначити, наскільки добре MEGA впорається з моделями послідовностей у контексті аудіо мовлення, що дозволить отримати більш детальну оцінку його можливостей та потенційних обмежень.

2.1. Механізм уваги

Традиційний механізм самоуваги (англ. self-attention) [76] є наступною функцією:

$$Y = \text{Attention}(X) = f(\frac{QK^T}{\sqrt{d}})V, \tag{2.1.1}$$

де $X = (x_1, ..., x_n)$ є вхідною послідовністю довжиною n, Attention : $\mathbb{R}^{n \times d} \to \mathbb{R}^{nd}$ є функцією самоуваги, а d є розмірністю входу. Також передбачається, що вхідні та вихідні послідовності мають однакову довжину.

$$Q = XW_q + b_q,$$

$$K = XW_k + b_k,$$

$$V = XW_v + b_v$$

є послідовностями запитів, ключів та значень, з навчуваними параметрами $W_q,\ W_k,\ W_v \in \mathbb{R}^{d\times d}$, та $b_q,\ b_k,\ b_v \in \mathbb{R}^d$. $f(\cdot)$ є активаційною функцією, наприклад, функцією softmax.

Матриця $A = f(\frac{QK^T}{d_k}) \in \mathbb{R}^{n\times n}$ називається *матрицею уваги*, оскільки вона визначає вагу сили залежності між кожною парою токенів у X. Оскільки вона моделює ваги парних залежностей, матриця A в принципі забезпечує гнучкий та потужний механізм для навчання залежностей на великій відстані з мінімальними індуктивними упередженнями. Однак на практиці це складне завдання - виявити всі шаблони взаємозв'язків у A безпосередньо з даних, особливо при роботі з довгими послідовностями. Крім того, обчислення A з h головами уваги займає $O(hn^2)$ простору та часу, і квадратична залежність від довжини послідовності стає значним обмеженням.

2.2. Механізм уваги з рухомим середнім та гейтом

Механізм гейтової уваги у Mega [2] використовує одиницю гейтової рекурентної одиниці (GRU) та гейтову увагу (GAU) [101] як основу. По-перше, обчислюється спільне представлення з використанням експоненційного рухомого середнього (англ. Exponential Moving Average, EMA)

$$X' = \text{EMA}(X) = \alpha \odot \mathbf{x}_t + (1-\alpha) \odot \mathbf{y}_{t-1} \tag{2.2.1}$$

$$Z = \phi_{\text{silu}}(X'W_z + b_z) \tag{2.2.2}$$

де X' є контекстуальним входом, а Z є спільним контекстом з розмірністю z, з проекційною матрицею $W_z \in \mathbb{R}^{d\times z}$ та зміщенням $b_z \in \mathbb{R}^z$.

EMA та механізми уваги мають свої обмеження, незважаючи на їх широке застосування та вражаючі успіхи у моделюванні послідовностей. Використовуючи їх властивості для доповнення один одного, EMA є частиною розрахунок матриці уваги A. Отримана модель користується перевагами сильного індуктивного упередження, зберігаючи при цьому здатність вивчати складні

патерни залежностей. Крім того, це поєднання дозволяє розробити обчислювально ефективний механізм уваги зі шматками з лінійною складністю відносно довжини послідовності.

Mega вводить модифікацію стандартного EMA, що називається *багатовимірний демпфований EMA*, для покращення його гнучкості та потужності.

Демпфований EMA. Попередні дослідження [102, 103] показали, що ослаблення з'єднаних ваг попередніх і поточних спостережень ($\boldsymbol{\alpha}$ проти $1-\boldsymbol{\alpha}$) забезпечує надійне моделювання залежностей. Натхненні цим, Mega дозволяє демпфування впливу попереднього кроку часу:

$$\mathbf{y}_t = \boldsymbol{\alpha} \odot \mathbf{x}_t + (1 - \boldsymbol{\alpha} \odot \boldsymbol{\delta})\mathbf{y}_{t-1}, \tag{2.2.3}$$

де $\boldsymbol{\delta} \in (0,1)^d$ є фактором демпфування.

Багатовимірний демпфований EMA. Для подальшого покращення виразності EMA ми вводимо багатовимірний варіант EMA. Конкретно, спочатку ми розширюємо кожен вимір вхідної послідовності $\boldsymbol{X}$ індивідуально до h вимірів за допомогою матриці розширення $\boldsymbol{\beta} \in \mathbb{R}^{d\times h}$. Формально, для кожного виміру $j \in \{1, 2, \ldots, d\}$:

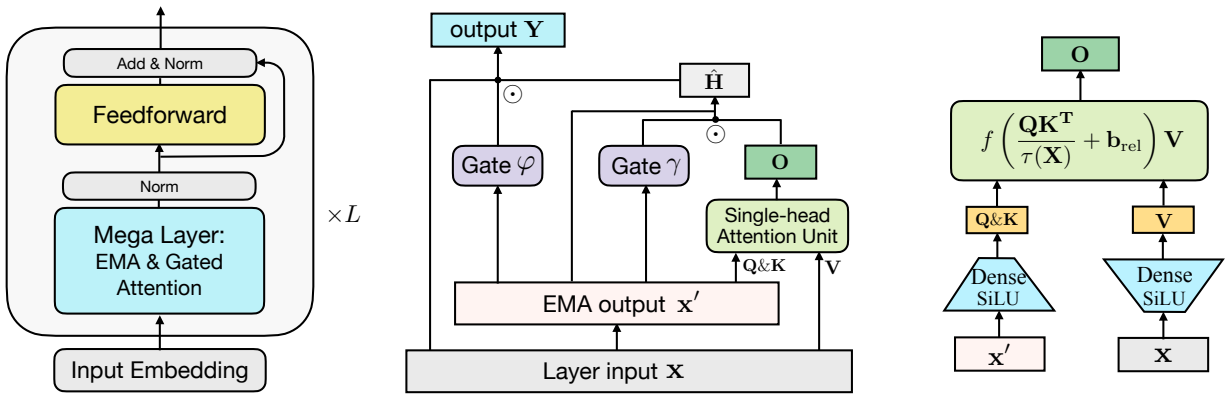
$$\mathbf{u}_t^{(j)} = \boldsymbol{\beta}_j \mathbf{x}_{t,j} \tag{2.2.4}$$

де $\boldsymbol{\beta}_j \in \mathbb{R}^h$ є j-им рядком $\boldsymbol{\beta}$, $\mathbf{u}_t^{(j)} \in \mathbb{R}^h$ є розширеним h-вимірним вектором для j-го виміру в момент часу t.

Відповідно, ми розширюємо форму $\boldsymbol{\alpha}$ та $\boldsymbol{\delta}$ з одномірного вектора до двовимірної матриці, тобто $\boldsymbol{\alpha}$, $\boldsymbol{\delta} \in \mathbb{R}^{d\times h}$, де $\boldsymbol{\alpha}_j$, $\boldsymbol{\delta}_j \in \mathbb{R}^h$ позначають j-ий рядок $\boldsymbol{\alpha}$ та $\boldsymbol{\delta}$ відповідно. Потім, для кожного виміру j, демпфований EMA застосовується до h-вимірного прихованого простору:

$$\begin{aligned} \mathbf{h}_t^{(j)} &= \boldsymbol{\alpha}_j \odot \mathbf{u}_t^{(j)} + (1 - \boldsymbol{\alpha}_j \odot \boldsymbol{\delta}_j) \odot \mathbf{h}_{t-1}^{(j)} \\ \mathbf{y}_{t,j} &= \boldsymbol{\eta}_j^T \mathbf{h}_t^{(j)} \end{aligned} \tag{2.2.5}$$

де $\mathbf{h}_t^{(j)} \in \mathbb{R}^h$ є прихованим станом EMA для j-го виміру в момент часу t. $\boldsymbol{\eta} \in \mathbb{R}^{d\times h}$ є проекційною матрицею для відображення h-вимірного прихова-



(a) Mega architecture. (b) Mega layer. (c) Single-head attention unit.

Рис. 2.1: Mega – графічне представлення архітектури [2]. Зліва (a) показано загальну архітектуру Mega блоку. По-центру (b) проілюстровано гейтований механізм уваги базуючись на EMA, а справа (c) проілюстрований single-head attention unit

ного стану назад до одномірного виходу $\mathbf{y}_{t,j} \in \mathbb{R}$. $\boldsymbol{\eta}_j \in \mathbb{R}^h$ є j-им рядком . Вихід $\boldsymbol{Y}$ з (2.2.5) позначається як $\boldsymbol{Y} \triangleq \mathrm{EMA}(\boldsymbol{X})$. Оскільки нам не потрібно явно обчислювати $\mathbf{h}_t^{(j)}$ для отримання виходу $\mathbf{y}_{t,j}$, часова та просторова складність схожа на стандартний EMA в (2.2.1). Експериментальні покращення демонструють його ефективність (§**??**).

Подібно до GAU, представлення запитів та ключів обчислюються за допомогою поелементних множників та зміщень до Z, а послідовність значень - з оригінального X:

$$Q = \kappa_q \odot Z + \mu_q \qquad \in \mathbb{R}^{n \times z} \tag{2.2.6}$$

$$K = \kappa_k \odot Z + \mu_k \qquad \in \mathbb{R}^{n \times z} \tag{2.2.7}$$

$$V = \phi_{\text{silu}}(XW_v + b_v) \qquad \in \mathbb{R}^{n \times v} \tag{2.2.8}$$

де κ_q, μ_q, κ_k, $\mu_k \in \mathbb{R}^z$ є навчуваними множниками та зміщеннями запитів та ключів відповідно. v є розширеним проміжним розміром для послідовності значень. Вихід уваги (англ. single head attention unit) обчислюється наступним

чином:

$$O = f\left(\frac{QK^T}{\tau(X)} + b_{\text{rel}}\right)V \qquad \in \mathbb{R}^{n\times v}. \tag{2.2.9}$$

де $\tau(X)$ є коефіцієнтом масштабування, який був встановлений до d_k.

У виразі термін $b_{\text{rel}} \in \mathbb{R}^{n\times n}$ є відносним позиційним упередженням.

Далі, Mega вводить гейти скидання γ та оновлення φ, і обчислює кандидатний вихід активації $\hat{H}$:

$$\begin{aligned}
\boldsymbol{\gamma} &= \phi_{\text{silu}}(\boldsymbol{X}'W_\gamma + b_\gamma) && \in \mathbb{R}^{n\times v}\\
\boldsymbol{\varphi} &= \phi_{\text{sigmoid}}(\boldsymbol{X}'W_\varphi + b_\varphi) && \in \mathbb{R}^{n\times d}\\
\hat{\boldsymbol{H}} &= \phi_{\text{silu}}(\boldsymbol{X}'W_h + (\boldsymbol{\gamma}\odot\boldsymbol{O})U_h + b_h) && \in \mathbb{R}^{n\times d}
\end{aligned}$$

Остаточний вихід Y обчислюється за допомогою гейту оновлення φ:

$$Y = \varphi \odot \hat{H} + (1-\varphi)\odot X \qquad \in \mathbb{R}^{n\times d} \tag{2.2.10}$$

Графічне представлення MEGA показано на рисунку 2.1

2.2.1. Дискретне аудіо представлення

Поширеним підходом є дискретизація зображення або аудіосигналу з використанням автоенкодера з векторною квантизацією, як у VQ-VAE [104]. Основна ідея полягає в тому, щоб відображати вихідні вектори енкодера на найближчий вектор з кодової книги e. Після цього відображені вектори кодової книги передаються декодеру. Ціль функції навчання має наступний вигляд:

$$L = \log p(x|z_q(x)) + \beta\|z_e(x) - sg[e]\|_2^2$$

де sg є оператором зупинки градієнта, який є тотожним при прямому проході і має нульовий градієнт, таким чином ефективно обмежуючи його параметр постійною змінною. Декодер оптимізує тільки перший член втрат, тоді як енкодер оптимізує перший і другий члени втрат. Архітектура автоенкодера показана на Рисунку 2.2.

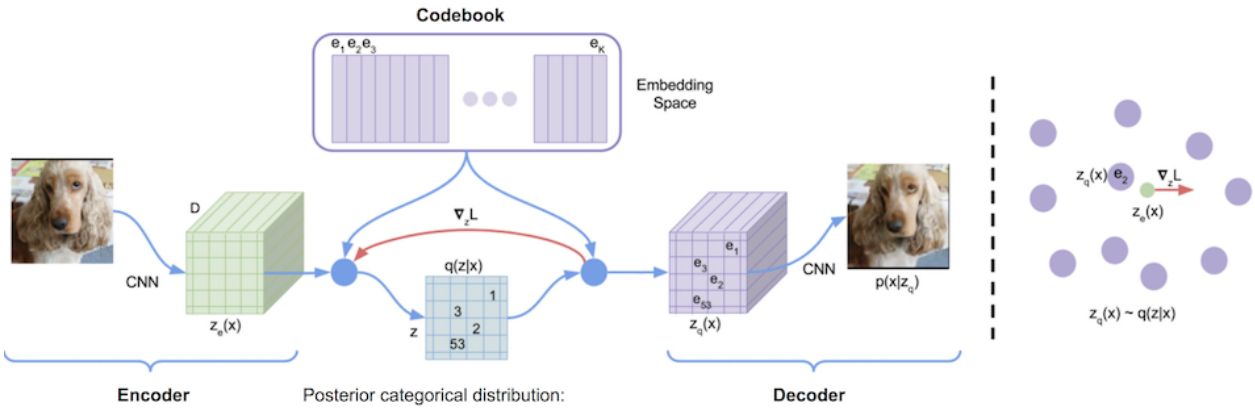


Рис. 2.2: VQ-Vae архітектура [104]

2.3. Чисельний експеримент

По-перше, модель VQ-VAE була попередньо навчена на датасеті LJ-Speech для отримання дискретного аудіо представлення. Вхідними даними були мел-спектрограми. Це призвело до латентного простору, який у 4 рази менший за початкову спектрограму. Він складається з одномірної послідовності дискретних точок, які є 512-вимірними векторами. Усього є 8192 дискретних вектора з кодової книги.

Потім дві авторегресійні моделі були навчені на цих латентних представленнях. Перша модель є традиційним декодером трансформера з каузальною самоувагою, що нагадує GPT [105]. Друга модель є трансформером з увагою, що використовує EMA. Обидві моделі були навчені для максимізації наступної цілі:

$$L = \sum_{x,y} \log P(y|x_1, ..., x_m).$$

Розміри моделей однакові, приблизно 23.5 мільйона параметрів, як і інші гіперпараметри. Криві втрат показані на Рисунку 2.3.

Як можна побачити, модель трансформера з традиційним механізмом уваги показує кращі результати на функції втрат, проте трансформер з увагою на основі EMA сходиться набагато швидше. Крім цього, EMA модель показує ознаки перенавчання.

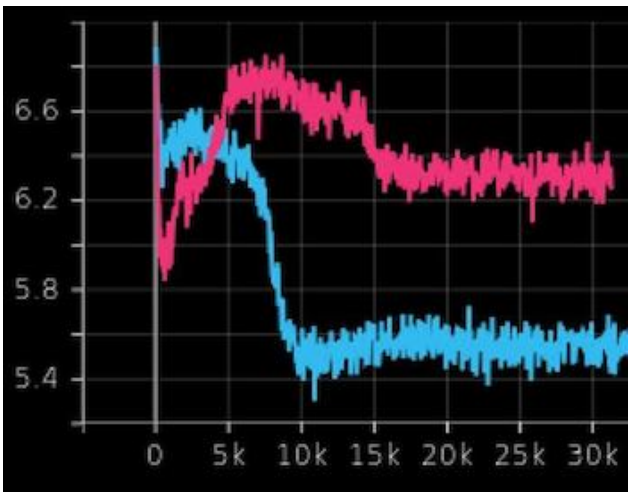


Рис. 2.3: Авторегресійні втрати на тренуванні. Рожева крива: EMA Gated Attention, синя крива: традиційний трансформер

2.4. Висновок

У цій роботі був проведений експеримент для порівняння різних механізмів уваги на дискретному мовному представленні. Можна зробити висновок, що традиційна самоувага показує кращі результати, хоча модель на основі уваги з використанням EMA сходиться набагато швидше. Це показує, що механізм уваги на основі EMA поки що не є достатньо надійним та стабільним, і, ймовірно, вимагає більш ретельного налаштування гіперпараметрів.

РОЗДІЛ 3
АНАЛІЗ ДИФУЗІЙНОГО МОДЕЛЮВАННЯ НА ПРИКЛАДІ АУДІО СИГНАЛІВ

3.1. Маскований автокодувальник для компресії ознак

Маскований автокодувальник (англ. Masked Auto-encoder, MAE) обробляє вхідний аудіосигнал x шляхом обчислення його логарифмічної мел-спектрограми $X \in \mathbb{R}^{T \times F}$, де T позначає часові кроки, а F представляє частотні смуги спектрограми. Ця спектрограма X аналогічна зображенню та розділяється на блоки розміром $P \times P$, де розмір кожного блоку P є дільником як T, так і F. Ці блоки потім подаються на вхід до енкодера AudioMAE. Енкодер, який є згортковою нейронною мережею, працює з ядром і кроком, встановленими на P, генеруючи вихід із D каналами. Таким чином, вихід енкодера $E \in \mathbb{R}^{T' \times F' \times D}$, де $T' = \frac{T}{P}$ і $F' = \frac{F}{P}$, а D є розмірністю векторів ознак, створених MAE. Закодовані ознаки E розглядаються як латентне представлення для подальшої обробки.

Для тренування AudioMAE використовується функція втрат, зокрема середньоквадратична похибка (MSE), обчислювана за маскованими блоками для оцінки якості реконструкції. Функція втрат MSE визначається як:

$$\text{MSE Loss} = \frac{1}{N_{\text{masked}}} \sum_{i=1}^{N_{\text{masked}}} \left(\hat{X}_i - X_i \right)^2, \tag{3.1.1}$$

де N_{masked} - кількість маскованих блоків, X_i - оригінальний блок, а $\hat{X}_i$ - реконструйований блок, виданий декодером MAE.

3.2. Інформація про контекст C: вхідне аудіо та текстові фонеми

В моделі генерації аудіо інформація про контекст C відіграє ключову роль у спрямуванні процесу генерації, надаючи контекстуальні підказки, що впливають на вихідні дані. Для цієї моделі C отримується з двох основних джерел: вхідне аудіо та текстові фонеми, кожне з яких робить унікальний внесок у процес генерації.

3.2.1. CLAP автокодувальник для встановлення контексту

CLAP автокодувальник [106] призначений для проекції як аудіо, так і тексту в єдиний мультимодальний простір, що дозволяє ефективно використовувати цю інформацію як дані для контексту. Нехай X_a позначає оброблене аудіо, представлене у вигляді матриці $X_a \in \mathbb{R}^{F \times T}$, де F - кількість спектральних компонентів, таких як мел-спектри, і T - кількість часових бінів. Аналогічно, нехай X_t позначає текстове представлення. У межах батчу N пар аудіо-текст ці дані позначаються як $\{X_a, X_t\}$.

Аудіо та текстові дані кодуються через окремі функції енкодера, $f_a(\cdot)$ та $f_t(\cdot)$, відповідно. Для батчу з N елементів закодовані представлення визначаються як:

$$\hat{X}_a = f_a(X_a); \quad \hat{X}_t = f_t(X_t) \tag{3.2.1}$$

де $\hat{X}_a \in \mathbb{R}^{N \times V}$ та $\hat{X}_t \in \mathbb{R}^{N \times U}$ представляють розмірності V та U аудіо та текстових представлень відповідно.

Для перенесення цих представлень у спільний мультимодальний простір розмірності d застосовуються навчальні лінійні проекції:

$$E_a = L_a(\hat{X}_a); E_t = L_t(\hat{X}_t) \tag{3.2.2}$$

де $E_a, E_t \in \mathbb{R}^{N \times d}$ є проекційними векторами для аудіо та тексту, а L_a, L_t є відповідними функціями лінійної проекції.

Схожість між аудіо та текстовими векторами обчислюється у спільному просторі наступним чином:

$$C = \tau \cdot (E_t \cdot E_a^\top) \tag{3.2.3}$$

де τ є температурним параметром, що масштабує діапазон вихідних векторів. Матриця схожості $C \in {}^{N \times N}$ включає правильні пари вздовж діагоналі та неправильні пари поза діагоналлю.

Симетрична функція втрат крос-ентропії обчислюється над матрицею схожості для тренування енкодерів та їхніх проекцій:

$$\mathcal{L} = 0.5 \cdot (\ell_{\text{text}}(C) + \ell_{\text{audio}}(C)) \tag{3.2.4}$$

де $\ell_k = \frac{1}{N} \sum_{i=0}^{N} \log \operatorname{diag}(\operatorname{softmax}(C))$ вздовж текстових та аудіо осей відповідно. Ця функція втрат сприяє спільному тренуванню аудіо та текстових енкодерів, покращуючи їх здатність ефективно кодувати релевантні ознаки для задач генерації аудіо.

3.2.2. Кодування текстових фонем

Текстові фонеми представляють інший важливий компонент контекстуальної інформації. Фонеми, найменші одиниці звуку в мові, вилучаються з вхідного тексту та кодуються для захоплення лінгвістичних нюансів та артикуляційних особливостей, необхідних для створення зв'язного та контекстно відповідного мовлення у аудіо сигналі. Цей процес кодування перетворює текстові дані на послідовність фонетичних представлень, C_{phonemes}, які потім використовуються для задання контексту генерації мовленнєвого сигналу, забезпечуючи відповідність створеного аудіо запланованому лінгвістичному змісту та стилю, заданому вхідним текстом.

Разом ці компоненти контексту $C = \{C_{\text{ref}}, C_{\text{phonemes}}\}$ інтегрують кілька модальностей - аудіо та текст, надаючи комплексний набір підказок, що по-

кращують здатність моделі створювати високоякісні та контекстно багаті аудіо виходи.

3.3. Авторегресивне моделювання для проміжного представлення

Цей компонент моделі відповідає за генерацію латентного представлення з різноманітної інформації про контекст за допомогою авторегресивного підходу, натхненного моделями на основі трансформерів. Формулювання авторегресивної моделі $\mathcal{M}_\theta$ задано як:

$$\hat{Y} = \mathcal{M}_\theta(C), \tag{3.3.1}$$

де C представляє інформацію про контекст, а $\hat{Y}$ є передбаченим латентним представленням. Модель $_\theta$, параметризована θ, прогнозує наступний елемент послідовності на основі попередніх, максимізуючи ймовірнісний розподіл по всій послідовності:

$$\text{argmax}_\theta \prod_{i=1}^{L} P(y_i | C_{\text{ref}}, C_{\text{phonemes}}, y_1, y_2, ..., y_{i-1}; \theta), \tag{3.3.2}$$

де L - це довжина латентної послідовності Y, закодованої MAE, а y_i - це її компоненти.

3.4. Варіаційний автокодувальник (VAE) для дифузійного моделювання

Варіаційний автокодувальник (VAE) [107] використовується головним чином для компресії ознак і навчання компактного аудіо представлення, z, яке є значно меншого розміру, ніж оригінальний аудіосигнал, x.

Операція VAE може бути виражена через рівняння:

$$\mathcal{V} : X \mapsto z \mapsto \hat{X} \tag{3.4.1}$$

де X представляє мел-спектрограму аудіо входу x, а $\hat{X}$ є реконструкцією X. Ця реконструйована спектрограма $\hat{X}$ може згодом бути перетворена назад

в аудіо хвильову форму $\hat{x}$ з використанням попередньо навченого вокодера HiFiGAN [108].

Для оптимізації параметрів VAE обчислюються функції втрат реконструкції та дискримінативна функція втрат на основі порівняння між X і $\hat{X}$. Крім того, архітектура VAE застосовує стратегію регуляризації шляхом обчислення розходження Кульбака-Лейблера (KL) між латентним представленням z і стандартним гаусівським розподілом з середнім $\mu = 0$ і дисперсією $\sigma^2 = 1$:

$$\text{KL Loss} = D_{\text{KL}}\left(\mathcal{N}(z; \mu_z, \sigma_z^2) \parallel (0, 1)\right) \tag{3.4.2}$$

Ця регуляризація допомагає підтримувати статистичні властивості латентного простору, забезпечуючи, що z дотримується гаусівського розподілу, тим самим стабілізуючи процес генерації та підвищуючи якість реконструйованого аудіо.

3.5. Модель латентної дифузії для синтезу аудіо

Синтез аудіо виконується за допомогою моделі латентної дифузії, яка працює у латентному просторі, наданому VAE автокодувальником. Ця модель виражена через серію кроків дифузії, починаючи з латентного представлення z і поступово додаючи шум до досягнення стану дифузії z_T:

$$z_t = \sqrt{1 - \beta_t} z_{t-1} + \sqrt{\beta_t} \epsilon_t, \tag{3.5.1}$$

де β_t - це параметр графіку шуму, а $\epsilon_t \sim \mathcal{N}(0, I)$ - це гаусівський шум.

Зворотний процес включає поступове видалення шуму з z_T для реконструкції латентного представлення:

$$z_{t-1} = \frac{z_t - \sqrt{\beta_t} \epsilon_t}{\sqrt{1 - \beta_t}}. \tag{3.5.2}$$

Оптимізація спрямована на мінімізацію різниці між оригінальними та реконструйованими латентними представленнями, визначеними функцією

втрат:

$$\mathcal{L}(\phi) = \mathbb{E}_{z_0,\epsilon\sim\mathcal{N}(0,I),t}\left[\|z_0 - \mathrm{Dec}(z_t;\phi)\|^2\right], \tag{3.5.3}$$

де ϕ - це параметри дифузійної моделі, Dec позначає функцію декодування дифузійної моделі, а z_0 - це оригінальне латентне представлення.

3.5.1. Адаптація компонентів попередньо навченої моделі

У розробці моделі в даній роботі були використані компоненти попередньо навченої моделі Audio Latent Diffusion Model 2 (Audio LDM2) [109]. Цей підхід дозволив скористатися міцними основами, закладеними існуючою моделлю, зокрема її ефективною обробкою складних аудіо даних через дифузійні процеси. Важливою модифікацією в методології стала адаптація механізму задання контексту, що використовується в Audio LDM2.

Традиційно, Audio LDM2 використовує контекстуальний вектор C, який включає текстові векторні представлення, закодовані CLAP, для керування процесом синтезу аудіо. На противагу цьому, запропонована модель замінює ці текстові представлення на аудіо-векторні представлення, закодовані CLAP, що мають намір кодувати емоції та інформацію про мовця. Ця зміна краще узгоджується з фокусом у даній роботі на покращенні якості аудіо та релевантності у додатках тексту в мовлення, де пряма кореляція між характеристиками вхідного аудіо та згенерованим виходом є вирішальною.

$$C_{\text{ref}} = f_a(X_{ref}), \tag{3.5.4}$$

де C_{ref} представляє новий контекстуальний вектор, використовуючи аудіо представлення, $f_a(\cdot)$ - це CLAP аудіо енкодер, а X_{ref} - це матриця ознак довідкового аудіо.

Ця адаптація не тільки налаштовує модель більш точно до специфічного випадку використання у даній роботі, але й оптимізує взаємодію між інформацією про контекст та генеративними компонентами моделі. Інтегрую-

чи аудіо представлення безпосередньо, дана модель набуває більш детального розуміння аудіо ознак, що може привести до більш точної та природної генерації аудіо в системах TTS.

3.6. Оцінювання дифузійної моделі

3.6.1. Метрика схожості голосу

Для кількісної оцінки схожості голосу між референтними та згенерованими аудіозаписами було використано модель верифікації голосу на основі WavLM, сучасної моделі обробки аудіо [110]. Ця модель була попередньо навчена з використанням контрастної loss-функції, яка оптимізує вектоии шляхом мінімізації відстані між схожими парами та максимізації відстані для несхожих пар, що робить її придатною для задач верифікації голосу.

Метрика, яка наводиться, — це середня косинусна відстань між векторами-ембедінгами референтних аудіозаписів та відповідних згенерованих зразків. Вектори-ембедінги отримуються за допомогою моделі WavLM, яка кодує характеристики, специфічні для голосу. Косинусна відстань вимірює косинус кута між двома векторами у просторі ембедінгів, надаючи шкалу від -1 (повністю різні) до 1 (ідентичні), де вищі значення вказують на більшу схожість голосу. Формула для косинусної відстані має вигляд:

$$\text{Косинусна відстань} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.6.1}$$

де A_i та B_i є компонентами векторів-ембедінгів з референтних та згенерованих аудіозаписів відповідно. Ця метрика кількісно оцінює, наскільки добре згенероване аудіо зберігає характеристики голосу референтного аудіо. Результати представлені у Таблиці 3.1.

Модель	**Speaker similarity ↑**
Запропонована модель	0.63
xTTS v2	0.90

Таблиця 3.1

Порівняння оцінок схожості голосу.

3.6.2. Точність класифікації емоцій

Точність розпізнавання емоцій оцінювалася за допомогою моделі Emotion2Vec [111], яка прогнозувала емоції для як референтних, так і згенерованих аудіо. Ця міра відображає здатність моделі кодувати та відтворювати емоційні стани, заплановані в оригінальній мові. Результати представлені у Таблиці 3.2.

Модель	**Точність класифікації емоцій ↑**
Запропонована модель	0.035
xTTS v2	0.17

Таблиця 3.2

Порівняння оцінок точності класифікації емоцій.

3.6.3. Словесна та символьна похибка

Ці метрики були обчислені за допомогою транскриптів, згенерованих попередньо навченою великою моделлю автоматичного розпізнавання мови Whisper [112], порівнюючи їх з еталонними транскриптами.

Словесна похибка обчислюється як відношення загальної кількості операцій (вставок, видалень та замін), необхідних для перетворення згенерованого транскрипту в еталонний, до загальної кількості слів в еталонному транскрипті. Іншими словами, дана похибка обчислюється як відстань Ле-

венштейна. Формула для WER має вигляд:

$$\text{WER} = \frac{S + D + I}{N} \tag{3.6.2}$$

де S — кількість замін, D — кількість видалень, I — кількість вставок, а N — кількість слів в еталонному транскрипті.

Аналогічно, символьна похибка обчислюється за тим же принципом на рівні символів, а не слів. Вона вимірює мінімальну кількість вставок, видалень та замін, необхідних для зміни згенерованого транскрипту в еталонний, нормалізовану за загальною кількістю символів в еталонному транскрипті. Формула для CER:

$$\text{CER} = \frac{s + d + i}{n} \tag{3.6.3}$$

де s представляє заміни, d представляє видалення, i представляє вставки, а n — загальна кількість символів в еталонному транскрипті.

Обидві метрики надають важливі інсайти щодо точності транскрипції згенерованої мови, де нижчі значення вказують на вищу точність та кращу продуктивність системи синтезу тексту в мову. Результати представлені у Таблиці 3.3.

Модель	**WER ↓**	**CER ↓**
дифузійна модель	1.0	1.01
xTTS v2	0.21	0.02

Таблиця 3.3

Порівняння словесної похибки (WER) та символьної похибки (CER).

3.7. Висновки до розділу

Це дослідження надало оцінку латентної дифузійної моделі у порівнянні з моделлю xTTS v2, використовуючи набір строгих метрик на наборі даних EmoV-DB. Висновки виявили деякі інсайти щодо продуктивності обох моделей з точки зору схожості голосу, збереження емоцій та зрозумілості.

Хоча запропонована модель продемонструвала пристойну продуктивність у збереженні характеристик голосу, як свідчить оцінка схожості голосу, вона поступалася xTTS v2 за всіма оціненими метриками. Зокрема, запропонована модель виявила значні недоліки в точності класифікації емоцій, що свідчить про те, що ембедінги CLAP аудіо, на які вона спирається, можуть бути більш схильні для кодування інформації, пов'язаної з голосом, ніж нюансів емоційного вираження. Це спостереження було підкреслено низьким рівнем похибки класифікації емоцій, що вказує на можливу невідповідність між емоційними намірами, закодованими в векторах-ембедінгах, і тими, що виражені в мовному виході. Також, тренувальний набір даних для компонента CLAP, який є сумішшю мовлення та загального аудіо та відповідних описів, може бути неефективним для синтезу мовлення, що вказує на те, що попереднє навчання на транскрибованому наборі даних мовлення може покращити якість генерації.

Висока продуктивність xTTS v2 у всіх аспектах свідчить про те, що її архітектура моделі або навчальний режим можуть краще інтегрувати та балансувати як інформацію про голос, так і емоції. Це підкреслює критичну область для майбутнього покращення для AudioLDM2, вказуючи на те, що подальше вдосконалення здатності енкодера обробляти та інтегрувати емоційні дані може підвищити її продуктивність.

На завершення, результати цього дослідження свідчать про те, що, хоча використання аудіо-базованих ембедінгів CLAP як підказки для авторегресійної моделі пропонує перспективний напрямок для покращення розбірливості голосу в синтезованому мовленні, залишається значний простір для покращення точного захоплення та відтворення емоційних нюансів. Майбутні дослідження повинні зосередитися на оптимізації балансу між характеристиками голосу та емоціями в моделях TTS для досягнення більш цілісного та ефективного синтезу людського мовлення.

РОЗДІЛ 4
ПАРАМЕТРИЗАЦІЯ І ВИГЛЯД ФУНКЦІЙ АКТИВАЦІЇ

Функції активації є ключовим компонентом в архітектурі нейронної мережі, істотно впливаючи на продуктивність моделі та ефективність навчання. Попри те, що традиційні одновимірні функції активації були широко вивчені і оптимізовані, дослідження багатовимірних функцій активації залишається відносно малодослідженим в літературі. Вмотивована цією прогалиною, ця робота спрямована на оцінку та потенціал багатоваріантних функцій активації в різних областях і завданнях. У цій роботі функції активації класифіковані на основі двох основних критеріїв: навчальні параметри і розмірність.

4.1. Огляд і класифікація функцій активації

4.1.1. Одномірні активаційні функції без ннавчальних параметрів

Сигмоїда Функція сигмоїда визначається як:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.1.1}$$

Вона відображає будь-яке дійсне число в діапазон (0, 1), що робить її придатною для задач бінарної класифікації. Плавний градієнт функції сигмоїда є перевагою для методів оптимізації на основі градієнтів. Однак функція сигмоїда страждає від проблеми зникаючого градієнта для великих позитивних або негативних входів, що може ускладнити навчання глибоких мереж. Крім того, вихід функції сигмоїда не є нуль-центрованим, що може спричинити проблеми з градієнтами, особливо у глибших мережах.

Tanh. Функція гіперболічного тангенса (tanh) визначається як:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.1.2}$$

Вона відображає вхідні значення в діапазон (-1, 1) і часто використовується в прихованих шарах нейронних мереж. Функція tanh має перевагу нульцентрованих виходів, що може сприяти швидшій збіжності під час навчання. Подібно до функції сигмоїда, tanh також має плавний градієнт, що є корисним для оптимізації. Однак вона також страждає від проблеми зникаючого градієнта для великих вхідних значень, що може уповільнити навчання глибоких нейронних мереж.

Rectified linear units (ReLU) Випрямляючі лінійні одиниці (ReLU) є сімейством функцій, що включають ReLU, Leaky ReLU, PReLU та ELU.

ReLU: Випрямлена лінійна одиниця (ReLU) визначається як:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{4.1.3}$$

ReLU широко використовується через свою простоту та обчислювальну ефективність. Вона ефективно пом'якшує проблему зникаючого градієнта, що є поширеною в сигмоїдних та tanh функціях. Однак, ReLU може страждати від проблеми "мертвих ReLU коли нейрони можуть стати неактивними та видавати лише нулі.

Leaky ReLU: Leaky ReLU [113] вводить невеликий нахил для від'ємних значень, визначений як:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x \geqslant 0 \\ \alpha x & \text{if } x < 0 \end{cases} \tag{4.1.4}$$

де α - невелика константа (зазвичай 0.01). Leaky ReLU вирішує проблему "мертвих"ReLU, дозволяючи невеликий, ненульовий градієнт, коли одиниця не активна, забезпечуючи, щоб нейрони не ставали неактивними.

PReLU: Параметрична випрямлена лінійна одиниця (PReLU) [114] узагальнює Leaky ReLU, роблячи нахил α навчувальним або попередньо-

заданим параметром:

$$\mathrm{PReLU}(x) = \begin{cases} x & \text{if } x \geqslant 0 \\ \alpha x & \text{if } x < 0 \end{cases} \tag{4.1.5}$$

Ця гнучкість дозволяє моделі навчати оптимальний нахил під час тренування, що потенційно покращує продуктивність. Однак це вводить додаткові параметри, які потрібно навчати, збільшуючи складність моделі.

ELU: Експоненціальна лінійна одиниця (ELU) [115] визначається як:

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x \geqslant 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases} \tag{4.1.6}$$

де α - гіперпараметр, що контролює значення, до якого насичується ELU для від'ємних входів. ELU прагне наблизити середнє значення активації до нуля, що прискорює навчання. Вона також допомагає пом'якшити проблему зникаючого градієнта, але за рахунок збільшення обчислювальної складності порівняно з ReLU.

GELU. Гаусова похибкова лінійна одиниця (GELU) [116] визначається як:

$$\mathrm{GELU}(x) = x \cdot \Phi(x) \tag{4.1.7}$$

де $\Phi(x)$ - функція накопичувального розподілу стандартного нормального розподілу. GELU поєднує властивості ReLU та dropout, застосовуючи стохастичну регуляризацію, що може покращити узагальнення. Однак вона є більш обчислювально інтенсивною через участь похибкової функції.

Swish. Функція Swish [117] визначається як:

$$\mathrm{Swish}(x) = x \cdot \sigma(x) \tag{4.1.8}$$

де $\sigma(x)$ - функція сигмоїда. Swish є плавною, немонотонною функцією, яка показала кращі результати порівняно з ReLU на глибоких мережах. Властивість самогейтінгу Swish дозволяє їй зберігати деяку інформацію з від'ємних

входів, що потенційно призводить до кращого потоку градієнтів. Однак обчислення Swish є складнішим, ніж ReLU, що може уповільнити тренування.

Mish. Функція Mish [118] визначається як:

$$\text{Mish}(x) = x \cdot \tanh(\ln(1 + e^x)) \tag{4.1.9}$$

Mish є новою активаційною функцією, яка поєднує властивості як ReLU, так і Swish, показуючи перспективні емпіричні результати. Вона забезпечує плавні градієнти і є немонотонною, що може покращити динаміку навчання глибоких мереж. Однак, як і Swish, Mish є більш обчислювально складною, ніж ReLU.

Періодичні активаційні функції.

Періодичні активаційні функції [119] вводять індуктивні упередження, які забезпечують глобальну стаціонарність в нейронних мережах. Ці функції встановлюють зв'язок між апріорним розподілом ваг мережі та спектральною щільністю ковариаційної функції граничного стаціонарного гауссового процесу (GP) однорівневих баєсовських нейронних мереж (BNNs). Цей зв'язок виходить за межі синусоїдальних (Фур'є) активацій і включає інші періодичні функції, такі як трикутна хвиля і періодичні активаційні функції ReLU.

Загальна форма періодичної активаційної функції $\sigma(x)$ може бути описана як:

$$\sigma_{\text{periodic}}(x) = \sum_{k=1}^{\infty} a_k \sin(kx) + b_k \cos(kx),$$

де a_k і b_k є коефіцієнтами, що визначають конкретну форму періодичної функції. Ці функції є неперервними, обмеженими і центрованими на нулі, що забезпечує відсутність надмірно впевнених прогнозів за межами тренувальних даних.

Синусоїдна активаційна функція:

$$\sigma_{\text{sin}}(x) = \sqrt{2}\sin(x)$$

Синусо-косинусна активаційна функція:

$$\sigma_{\text{sin-cos}}(x) = \sin(x) + \cos(x)$$

Трикутна хвильова активаційна функція:

$$\sigma_{\text{triangle}}(x) = \frac{4}{\pi^2}\sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)^2}\sin((2k+1)x)$$

Періодична активаційна функція ReLU:

$$\sigma_{\text{pReLU}}(x) = \max(0, \sin(x)) + \max(0, \cos(x))$$

Періодичні активаційні функції показали порівнянну продуктивність на внутрішніх даних, при цьому значно покращуючи виявлення позапланових зразків шляхом повернення до апріорного розподілу, таким чином підвищуючи надійність та оцінку невизначеності глибоких нейронних мереж.

4.1.2. Існуючі багатовимірні активаційні функції

Гейтові лінійні одиниці (GLUs). Гейтові лінійні одиниці (GLUs) [120] вводять навчувальні гейти, що модулюють вхід. Різні варіанти GLU показали значні покращення в архітектурах трансформерів.

SwiGLU: Гейтові лінійні одиниці Swish (SwiGLU) визначаються як:

$$\text{SwiGLU}(x, W, V, b, c, \beta) = \text{Swish}_\beta(xW + b) \odot (xV + c) \tag{4.1.10}$$

де W і b є навчувальними вагами і зміщеннями, а $\odot$ позначає поелементне множення. SwiGLU використовує переваги активації Swish у гейтовому механізмі, покращуючи продуктивність у моделях трансформерів. Однак збільшення кількості параметрів і складність Swish можуть призвести до вищих обчислювальних витрат.

GeGLU: Гейтові лінійні одиниці GELU (GeGLU) визначаються як:

$$\text{GEGLU}(x, W, V, b, c) = \text{GELU}(xW + b) \odot (xV + c) \tag{4.1.11}$$

GeGLU інтегрує активаційну функцію GELU у гейтовий механізм, потенційно покращуючи продуктивність моделі через стохастичну регуляризацію та покращений потік градієнтів. Компроміс подібний до SwiGLU, з підвищеною обчислювальною складністю.

ReGLU: Гейтові лінійні одиниці ReLU (ReGLU) визначаються як:

$$\text{ReGLU}(x, W, V, b, c) = \max(0, xW + b) \odot (xV + c) \tag{4.1.12}$$

ReGLU поєднує простоту і ефективність ReLU з гейтовим механізмом, забезпечуючи баланс між продуктивністю та обчислювальними витратами. Хоча він може не забезпечувати такого ж приросту продуктивності, як SwiGLU або GeGLU, він залишається обчислювально ефективнішим.

Гейтові лінійні одиниці (GLUs) та їх варіанти показали значні покращення в різних архітектурах нейронних мереж, особливо в моделях трансформерів. Відмінною рисою варіантів GLU є їх здатність модулювати вхід через навчувальні гейтові механізми, що підвищує виразну здатність моделі та динаміку навчання. При розширенні на багатовимірні входи, варіанти GLU працюють, розділяючи вхідний вектор до активації на два окремих вектори, пропускаючи кожен через різні функції, а потім обчислюючи їх поелементний добуток.

Розглянемо вхідний вектор до активації $\mathbf{x} \in \mathbb{R}^d$. У контексті варіантів GLU цей вхідний вектор розділяється на два підвектори, $\mathbf{x}_1$ та $\mathbf{x}_2$, кожен з яких має розмірність $\frac{d}{2}$:

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$$

Кожен підвектор $\mathbf{x}_1$ і $\mathbf{x}_2$ потім пропускається через різні функції та гейтовий механізм. Для загального варіанту GLU обчислення можна описати

наступним чином:

$$\text{GLU}(\mathbf{x}) = \sigma(W_1\mathbf{x}_1 + b_1) \odot f(W_2\mathbf{x}_2 + b_2)$$

Тут W_1 та W_2 є ваговими матрицями, b_1 та b_2 є зміщеннями, σ є гейтовою функцією (часто сигмоїдна функція), а f є активаційною функцією, специфічною для варіанту GLU. Оператор $\odot$ позначає поелементний добуток.

Наприклад, у варіанті SwiGLU, активаційна функція f є функцією Swish:

$$\text{SwiGLU}(\mathbf{x}) = \sigma(W_1\mathbf{x}_1 + b_1) \odot \text{Swish}(W_2\mathbf{x}_2 + b_2)$$

Функція Swish визначається як:

$$\text{Swish}(z) = z \cdot \sigma(z)$$

Таким чином, для SwiGLU загальне обчислення стає:

$$\text{SwiGLU}(\mathbf{x}) = \sigma(W_1\mathbf{x}_1 + b_1) \odot ((W_2\mathbf{x}_2 + b_2) \cdot \sigma(W_2\mathbf{x}_2 + b_2))$$

Аналогічно, у варіанті GeGLU, активаційна функція f є функцією GELU:

$$\text{GeGLU}(\mathbf{x}) = \sigma(W_1\mathbf{x}_1 + b_1) \odot \text{GELU}(W_2\mathbf{x}_2 + b_2)$$

Функція GELU визначається як:

$$\text{GELU}(z) = z \cdot \Phi(z),$$

де $\Phi(z)$ є функцією накопичувального розподілу стандартного нормального розподілу. Таким чином, для GeGLU обчислення є:

$$\text{GeGLU}(\mathbf{x}) = \sigma(W_1\mathbf{x}_1 + b_1) \odot ((W_2\mathbf{x}_2 + b_2) \cdot \Phi(W_2\mathbf{x}_2 + b_2))$$

У ReGLU, активаційна функція f є функцією ReLU:

$$\text{ReGLU}(\mathbf{x}) = \sigma(W_1\mathbf{x}_1 + b_1) \odot \text{ReLU}(W_2\mathbf{x}_2 + b_2)$$

Функція ReLU визначається як:

$$\text{ReLU}(z) = \max(0, z)$$

Таким чином, для ReGLU обчислення стає:

$$\text{ReGLU}(\mathbf{x}) = \sigma(W_1\mathbf{x}_1 + b_1) \odot \max(0, W_2\mathbf{x}_2 + b_2)$$

Варіанти GLU, фактично, є біваріантними функціями та забезпечують надійний механізм для модуляції вхідних даних через навчувальні гейти. Розділяючи вхід до активації на два вектори та обробляючи їх через різні функції, варіанти GLU можуть захоплювати складніші взаємодії всередині даних, покращуючи здатність моделі до навчання складних шаблонів. Цей підхід не лише покращує продуктивність нейронних мереж у різних задачах, але й пропонує гнучку основу для інтеграції різних активаційних функцій та гейтових механізмів.

4.1.3. Активаційні функції з навчальними параметрами

ACON: Activate or Not.

У статті “Activate or Not: Learning Customized Activation” автори вводять нову активаційну функцію під назвою ACON (Activate or Not) [121], яка навчається активувати нейрони або ні на основі вхідних даних. Активаційна функція ACON може розглядатися як узагальнення і розширення існуючих активаційних функцій, таких як ReLU і Swish.

ACON функціонує шляхом введення коефіцієнта перемикання β, який контролює, чи буде активація поводитися лінійно або нелінійно. Активаційна функція ACON задається як:

$$\text{ACON}(x) = \sigma(\beta(x - p)) \cdot x + (1 - \sigma(\beta(x - p))) \cdot px,$$

де σ є сигмоїдною функцією, β є навчувальним коефіцієнтом перемикання, і p є навчувальним параметром. Параметр p дозволяє функції адаптивно переключатися між різними поведінками.

Автори пропонують три варіанти активаційної функції ACON:

- **ACON-A (Swish)**: Цей варіант використовує ту ж форму, що і активаційна функція Swish, і може бути записаний як:

  $$\text{ACON-A}(x) = x \cdot \sigma(\beta x),$$

  де $\sigma(\beta x)$ модулює вхід на основі навчувального параметра β.
- **ACON-B**: Цей варіант вводить додатковий параметр p, що дозволяє більш гнучко налаштовувати активацію:

  $$\text{ACON-B}(x) = (1 - p) \cdot x \cdot \sigma(\beta(1 - p)x) + p \cdot x$$

- **ACON-C**: Найбільш загальна форма, де як β, так і p є навчувальними параметрами, визначається як:

  $$\text{ACON-C}(x) = (p_1 - p_2) \cdot x \cdot \sigma(\beta(p_1 - p_2)x) + p_2 \cdot x$$

  Цей варіант дозволяє різне масштабування входу і надає додаткову гнучкість у навчанні оптимальної активаційної функції для даного завдання.

Сімейство активаційних функцій ACON динамічно регулює ступінь нелінійності під час навчання, забезпечуючи гладку і диференційовану альтернативу традиційним активаційним функціям. Ця гнучкість допомагає покращити узагальнення та продуктивність нейронних мереж у різних завданнях, включаючи класифікацію зображень, виявлення об'єктів та семантичну сегментацію. Завдяки навчанню оптимальної активаційної функції, ACON уникає необхідності вручну вибирати і налаштовувати активаційні функції для різних архітектур і наборів даних.

Активаційні одиниці Padé (PAUs)

Продуктивність навчання глибоких мереж значною мірою залежить від вибору нелінійної активаційної функції, асоційованої з кожним нейроном. Традиційні активаційні функції, такі як ReLU, сигмоїд і tanh, зазвичай фіксовані і накладають специфічні індуктивні упередження на мережу. Однак вибір оптимальної активаційної функції для даної архітектури і набору даних може бути непростим і часто вимагає емпіричного налаштування. Щоб вирішити цю проблему, активаційні одиниці Padé (PAUs) [122] вводять гнучкі параметричні раціональні функції, які можуть навчатися з кінця в кінець, усуваючи необхідність попереднього вибору фіксованих активаційних функцій.

Наближення Padé є ”найкращим”наближенням функції $f(x)$ раціональною функцією заданих порядків m і n. Математично, наближення Padé $F(x)$ представляється як відношення двох многочленів $P(x)$ і $Q(x)$:

$$F(x) = \frac{P(x)}{Q(x)} = \frac{\sum_{j=0}^{m} a_j x^j}{1 + \sum_{k=1}^{n} b_k x^k} = \frac{a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m}{1 + b_1 x + b_2 x^2 + \cdots + b_n x^n}$$

де a_j і b_k є коефіцієнтами многочленів. Наближення Padé часто забезпечує краще наближення, ніж ряд Тейлора, і може працювати в випадках, коли ряд Тейлора не збігається.

PAUs використовують цю раціональну форму функції для створення гнучкої активаційної функції, яку можна оптимізувати за допомогою стандартних методів зворотного поширення. Дозволяючи коефіцієнтам a_j і b_k бути навчувальними параметрами, PAUs можуть адаптуватися до специфічних потреб мережі під час навчання. Ця гнучкість дозволяє PAUs наближати загальні активаційні функції, такі як ReLU, сигмоїд і tanh, а також навчати нові, специфічні для завдання активаційні функції.

Щоб забезпечити числову стабільність і уникнути невизначених значень через полюси в раціональній функції, PAUs накладають обмеження на мно-

гочлен знаменника $Q(x)$. Зокрема, вони накладають умову $Q(x) \geqslant 1$ для всіх x, забезпечуючи, щоб раціональна функція залишалася добре визначеною:

$$F(x) = \frac{P(x)}{Q(x)} = \frac{\sum_{j=0}^{m} a_j x^j}{1 + |\sum_{k=1}^{n} b_k x^k|}$$

Це обмеження запобігає наближенню знаменника до нуля і викликанню нестабільності під час навчання та інференсу.

Градієнти, необхідні для зворотного поширення, виводяться наступним чином:

$$\begin{aligned}
\frac{\partial F}{\partial x} &= \frac{\partial P(x)}{\partial x}\frac{1}{Q(x)} - \frac{\partial Q(x)}{\partial x}\frac{P(x)}{Q(x)^2}, \\
\frac{\partial F}{\partial a_j} &= \frac{x^j}{Q(x)}, \\
\frac{\partial F}{\partial b_k} &= -\frac{x^k P(x)}{Q(x)^2}\frac{A(X)}{|A(X)|},
\end{aligned}$$

де $\frac{\partial P(x)}{\partial x}$ і $\frac{\partial Q(x)}{\partial x}$ є похідними многочленів по x, а $A(X)$ є членом $b_1 x + b_2 x^2 + +b_n x^n$.

Емпіричні оцінки показали, що PAUs можуть покращити прогнозну продуктивність на різних наборах даних і архітектурах. Завдяки навчанню оптимальної активаційної функції для кожного шару, PAUs надають універсальну і потужну альтернативу традиційним фіксованим активаційним функціям. Ця адаптивність також відкриває шлях для подальших досліджень у напрямку створення надій

4.2. Багатовимірні функції активації

У цьому розділі описуються багатовимірні активаційні функції, розглянуті в даному дослідженні, а саме: Gaussian Mixture Models (GMM), інтерполяційні активаційні функці та раціональні активаційні функції. Ці функції спрямовані на захоплення складних взаємодій у вхідному просторі, пропонуючи потенційні покращення продуктивності нейронних мереж.

4.2.1. Gaussian Mixture Models

Підхід Gaussian Mixture Models є ймовірнісною моделлю, яка припускає, що дані генеруються зі суміші декількох гауссових розподілів з невідомими параметрами. Коли вони використовуються як активаційні функції, активації на основі GMM визначаються як зважена сума гауссових компонентів, де як середні, так і дисперсії є навченими. Активаційна функція на основі GMM може бути виражена як:

$$\mathrm{GMM}(x) = \sum_{i=1}^{K} \pi_i \mathcal{N}(x; \mu_i, \Sigma_i),$$

де π_i — ваги суміші, μ_i — середні значення, а Σ_i — ковариаційні матриці гауссових компонентів. Ці параметри вивчаються під час навчання. Активаційні функції GMM можуть моделювати складніші вхідно-вихідні взаємозв'язки, але вводять значні обчислювальні витрати і вимагають ретельного налаштування для уникнення чисельної нестабільності.

4.2.2. Інтерполяційні функції активації

Активаційні функції на основі інтерполяції використовують навчальні параметри для інтерполяції між навчальними векторами, надаючи гнучкість і адаптивність у навчанні складних представлень. Ці методи спрямовані на захоплення складних шаблонів у даних шляхом динамічного налаштування активаційної функції під час навчання. Нехай $\mathbf{x} \in \mathbb{R}^{\text{batch\_size} \times \text{seq\_length} \times \text{dim}}$ є вхідним тензором перед активацією, де dim є розмірністю вхідних ознак. Функція використовує таблицю пошуку (LUT) з шириною 2, що призводить до $2^2 = 4$ точок інтерполяції. Навчені ваги позначаються як $\mathbf{W} \in \mathbb{R}^{\text{dim} \times 4}$.

Коефіцієнти інтерполяції обчислюються на основі зібраних вхідних даних:

$$\text{ratio}_0 = \frac{1 - x_0}{2},$$
$$\text{ratio}_1 = \frac{1 - x_1}{2},$$

де x_0 і x_1 є частинами вхідних даних перед активацією.

Навчені ваги з LUT розширюються, щоб відповідати розмірам пакету і послідовності. Нехай $\mathbf{W}_i$ позначає i-ий стовпець матриці ваг. Інтерпольовані значення b_0 і b_1 обчислюються як:

$$b_0 = \text{ratio}_1 \cdot \mathbf{W}_0 + (1 - \text{ratio}_1) \cdot \mathbf{W}_1$$

$$b_1 = \text{ratio}_1 \cdot \mathbf{W}_2 + (1 - \text{ratio}_1) \cdot \mathbf{W}_3$$

Кінцевий інтерпольований результат $\mathbf{r}$ потім обчислюється як:

$$\mathbf{r} = \text{ratio}_0 \cdot b_0 + (1 - \text{ratio}_0) \cdot b_1.$$

Для введення нелінійності застосовується механізм гейтів за допомогою сигмоїдної функції. Кінцевий вихід обчислюється як:

$$\mathbf{y} = \sigma(\mathbf{x}) \cdot (\mathbf{x} + \mathbf{r}),$$

де σ позначає сигмоїдну функцію. Це резидуальне з'єднання допомагає стабілізувати навчання і підвищити здатність моделі вивчати складні функції.

4.2.3. Багатовимірні раціональні активаційні функції (PAUs)

У цій роботі досліджуються раціональні активаційні функції (RAF) різних ступенів як потенційні кандидати для покращення продуктивності нейронних мереж. Ці $RAF2D$ функції використовують раціональні поліноми для моделювання складних вхідно-вихідних взаємозв'язків, з навчальними параметрами для чисельника і знаменника. Зокрема, було реалізовано RAF2D функції першого, другого і третього ступенів. Жорстко закодовані

параметри в цих функціях апроксимують активаційну функцію Swish-Gated Linear Unit (SwiGLU).

4.2.3.1. RAF2D першого ступеня

Активаційна функція RAF2D першого ступеня визначається як:

$$\mathrm{RAF2D}_1(x) = \frac{a_{00} + a_{01}x_2 + a_{10}x_1 + a_{11}x_1x_2}{1 + |b_{00} + b_{01}x_2 + b_{10}x_1 + b_{11}x_1x_2|},$$

де $x = [x_1, x_2]$ є вхідним тензором, поділеним на дві частини, і a_{ij} і b_{ij} є навчальними параметрами для чисельника і знаменника відповідно.

4.2.3.2. RAF2D другого ступеня

Активаційна функція RAF2D другого ступеня розширює складність шляхом включення квадратичних членів:

$$\mathrm{RAF2D}_2(x) = \frac{\sum_{i=0}^{2}\sum_{j=0}^{2} a_{ij}x_1^i x_2^j}{1 + \left|\sum_{i=0}^{2}\sum_{j=0}^{2} b_{ij}x_1^i x_2^j\right|},$$

де a_{ij} і b_{ij} є навчальними параметрами. Ця функція може захоплювати складніші взаємозв'язки між компонентами вхідних даних.

4.2.3.3. RAF2D третього ступеня

Активаційна функція RAF2D третього ступеня ще більше підвищує порядок полінома, дозволяючи ще складніші взаємодії:

$$\mathrm{RAF2D}_3(x) = \frac{\sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x_1^i x_2^j}{1 + \left|\sum_{i=0}^{3}\sum_{j=0}^{3} b_{ij}x_1^i x_2^j\right|},$$

де a_{ij} і b_{ij} є навчальними параметрами для чисельника і знаменника.

У реалізації ці активаційні функції RAF2D ініціалізуються параметрами для апроксимації активаційної функції Swish-Gated Linear Unit (SwiGLU).

4.3. Аналіз ефективності багатовимірних активаційних функцій

Для всебічного аналізу багатовимірних активаційних функцій розглядаються як задачі класифікації (моделювання мови та класифікація зображень), так і задачі регресії (моделювання латентної дифузії). Експерименти призначені для оцінки ефективності цих функцій активації в різних архітектурах нейронних мереж і наборах даних.

4.3.1. Об'єктивні метрики

У цьому розділі описуються метрики, які використовувалися для оцінювання результатів експериментів у різних завданнях, включаючи моделювання мови, класифікацію зображень та регресію в контексті моделювання латентної дифузії.

4.3.1.1. Метрики для моделювання мови

Perplexity

Перплексія (Perplexity) є основною метрикою для оцінки якості моделей мови. Вона вимірює, наскільки добре модель передбачає наступне слово у послідовності. Низька перплексія вказує на те, що модель добре передбачає наступні слова. Формула для обчислення перплексії:

$$\text{Perplexity}(P) = 2^{H(P)}$$

де $H(P)$ — це ентропія моделі.

4.3.1.2. Метрики для класифікації зображень

Точність

Точність (Accuracy) є основною метрикою для оцінки моделей класифікації зображень. Вона визначається як відсоток правильно класифікованих зразків серед усіх зразків. Формула для обчислення точності:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

4.3.1.3. Метрики для регресії (моделювання латентної дифузії)

Inception Score (IS)

Inception Score (IS) використовується для оцінки якості згенерованих зображень. Він базується на класифікаційних ймовірностях, отриманих від попередньо навченого Inception V3, і враховує як різноманітність, так і реалістичність зображень. Формула для обчислення IS:

$$\text{IS} = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g}\left[KL(p(y|\mathbf{x})||p(y))\right]\right)$$

де $p(y|\mathbf{x})$ — це умовний розподіл класів, а $p(y)$ — це маргінальний розподіл класів.

Fréchet Inception Distance (FID)

Fréchet Inception Distance (FID) оцінює якість згенерованих зображень, порівнюючи статистики реальних і згенерованих зображень у просторі ознак. Низькі значення FID вказують на те, що згенеровані зображення подібні до реальних. Формула для обчислення FID:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g})$$

де μ_r і μ_g — це вектори середніх значень реальних і згенерованих ознак відповідно, а Σ_r і Σ_g — це ковариаційні матриці ознак.

Spatial Fréchet Inception Distance (sFID)

Spatial Fréchet Inception Distance (sFID) є варіацією FID, яка враховує просторову інформацію у зображеннях, що дозволяє краще оцінити структурні аспекти зображень. Формула для обчислення sFID схожа на FID, але розраховується у просторово-залежному просторі ознак.

Precision і Recall

Precision і Recall оцінюють якість і різноманітність згенерованих зображень відповідно. Precision вимірює точність згенерованих зображень, а Recall оцінює, наскільки добре згенеровані зображення покривають різноманітність реальних зображень.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

де TP — кількість істинно позитивних зразків, FP — кількість хибно позитивних зразків, FN — кількість хибно негативних зразків.

Ці метрики забезпечують всебічне оцінювання моделей у різних завданнях і дозволяють зробити висновки про ефективність і узагальнювальну здатність різних активаційних функцій.

4.3.2. Результати моделювання мови

У цьому підрозділі оцінюється продуктивність багатовимірних функцій активації у контексті завдань моделювання мови в обробці природної мови (NLP). Була використана архітектура Transformer з 124 мільйонами параметрів, що складається з 12 шарів, прихованого розміру 768 та 12 голів уваги. Використовуються позиційні вектори-ембедінги з навчальними параметрами для захоплення позиційної інформації токенів. Для забезпечення ефективних експериментів була застосована бібліотека flash-attention.

Єдиною змінною у цих експериментах є функція активації. Було проведене порівняння кількох функцій активації, включаючи традиційні такі як ReLU та Swish, а також запропоновані багатовимірні функції активації. Метрики оцінювання включають значення функції втрат та перплексію на валідаційній вибірці даних, що забезпечує всебічну оцінку продуктивності моделі.

Таблиця 4.1

Значення функції втрат та перплексії для різних функцій активації після 3000 ітерацій оптимізатора.

Функція активації	**loss**	**perplexity↓**
$RAF2d_1$	3.649	38.418
$RAF2d_2$	3.719	41.235
$RAF2d_3$	3.774	43.544
GMM2d	3.81	44.221
Інтерполяційна ф-ція	3.648	38.405
ReLU	3.779	43.769
GeGLU	3.649	38.442
GELU	3.716	41.087
SwiGLU	**3.646**	**38.321**

Результати, представлені в таблиці 4.1, показують валідаційні втрати та валідаційну перплексію для кожної функції активації. Запропоновані багатовимірні функції активації, включаючи моделі Гауссової суміші (GMM), функції на основі інтерполяції та Padé Activation Units (PAU), демонструють кращу продуктивність порівняно з традиційними функціями активації, такими як ReLU та Swish.

Рисунки 4.1 ілюструють динаміку навчання моделі Transformer з різними функціями активації. Лівий графік показує втрати на тренуванні, центральний графік показує валідаційні втрати, а правий графік показує валідаційну перплексію. Ці графіки підкреслюють ефективність та дієвість багатовимірних функцій активації у покращенні процесу навчання моделі та загальної продуктивності.

4.3.3. Результати класифікації зображень

У цьому підрозділі оцінюється продуктивність багатовимірних функцій активації у контексті завдань класифікації зображень. Було використано архітектуру Vision Transformer [123] з 86 мільйонами параметрів, що складається з 12 шарів, прихованого розміру 768 та 12 голів уваги. Вхідне зображення обробляється у вигляді сітки з 16x16 патчів, кожен з яких проходить лінійну трансформацію, а потім перетворюється у 1-вимірну послідовність перед передачею до моделі трансформера. Оцінка проводиться на датасетах Imagenet та CIFAR-100.

Основною змінною в цих експериментах є функція активації. Було порівняно кілька функцій активації, включаючи традиційні такі як ReLU та GELU, а також запропоновані багатовимірні функції активації. Метрикою оцінювання є точність валідації після 40K оновлень оптимізатора, що надає чітке порівняння продуктивності моделі з різними функціями активації.

Таблиця 4.2 представляє точність валідації для кожної функції активації на датасетах Imagenet та CIFAR-100. Запропоновані багатовимірні функції активації, включаючи RAF2d з різними степенями, GMM2d та методи на основі інтерполяції, загалом перевершують традиційні функції активації такі як ReLU, GeGLU та GELU.

Рисунки 4.2 ілюструють динаміку навчання моделі Vision Transformer з різними функціями активації. Лівий графік показує втрати на тренуванні, центральний графік показує валідаційні втрати, а правий графік показує точність валідації. Ці графіки підкреслюють ефективність та дієвість багатовимірних функцій активації у покращенні продуктивності моделі на завданнях класифікації зображень.

Таблиця 4.2

Точність валідації після 40K ітерацій оптимізатора для різних функцій активації на наборі даних Imagenet. Рядки з відсутніми результатами означають що тренування було настабільним.

Функція активації	**loss**	**Accuracy**
$RAF2d_1$	1.87	57.52%
$RAF2d_2$	-	-
$RAF2d_3$	-	-
GMM2d	-	-
Інтерполяційна ф-ція	**1.854**	**57.93%**
ReLU	1.901	56.41%
GeGLU	2.196	50.35%
GELU	1.937	55.67%
SwiGLU	1.999	54.31%

4.3.4. Результати моделювання латентної дифузії

Для аналізу багатовимірних функцій активації для завдання регресії було обрано завдання моделювання латентної дифузії. Для цих експериментів були треновані модель дифузії з трансформерною архітектурою, що складається з 24 шарів, прихованого розміру 1024, розміром патчу 2 та 16 голів уваги. Навчання проводилося на датасеті ImageNet, використовуючи латентні представлення згорткової варіаційної автоенкодера (VAE). Зображення ImageNet були масштабовані до розміру 256x256 для навчання, а згортковий автоенкодер витягував латентні представлення розміром 32x32.

Ми оцінювали такі функції активації: SwiGLU, GELU та RAF2d-1-degree. Метрики продуктивності, використані для оцінки, включають Inception Score (IS), Fréchet Inception Distance (FID), sFID (spatial FID), Precision та Recall. Усі моделі були оцінені після приблизно 50K оновлень оптимізатора, що дає

уявлення про швидкість їх збіжності, хоча цей обсяг є занадто малим для таких датасетів як ImageNet.

Таблиця 4.3

Метрики продуктивності для різних функцій активації у моделюванні латентної дифузії.

Функція активації	**IS**	**FID**	**sFID**	**Precision**	**Recall**
SwiGLU	7.31	141.95	**36.81**	0.154	0.128
GELU	7.40	138.36	42.21	0.155	**0.168**
$RAF2d_1$	**8.0**	**138.05**	50.83	0.184	0.1
Інтерполяційна ф-ція	6.42	141.86	51.19	**0.189**	0.104

Таблиця 4.3 представляє результати оцінювання функцій активації у задачі регресії. Функція активації RAF2d-1-degree демонструє найкращу продуктивність за більшістю метрик, досягаючи найвищого Inception Score (IS), найнижчого Fréchet Inception Distance (FID) та найнижчого sFID. SwiGLU також показує хороші результати за всіма метриками. GELU, хоча й ефективний, показує трохи нижчу продуктивність порівняно з іншими двома функціями активації.

Ці результати підкреслюють ефективність багатовимірних функцій активації, особливо RAF2d-1-degree, у покращенні якості моделювання латентної дифузії. Використовуючи унікальні властивості цих функцій активації, можна досягти кращої генеративної продуктивності та більш точних представлень у моделях дифузії.

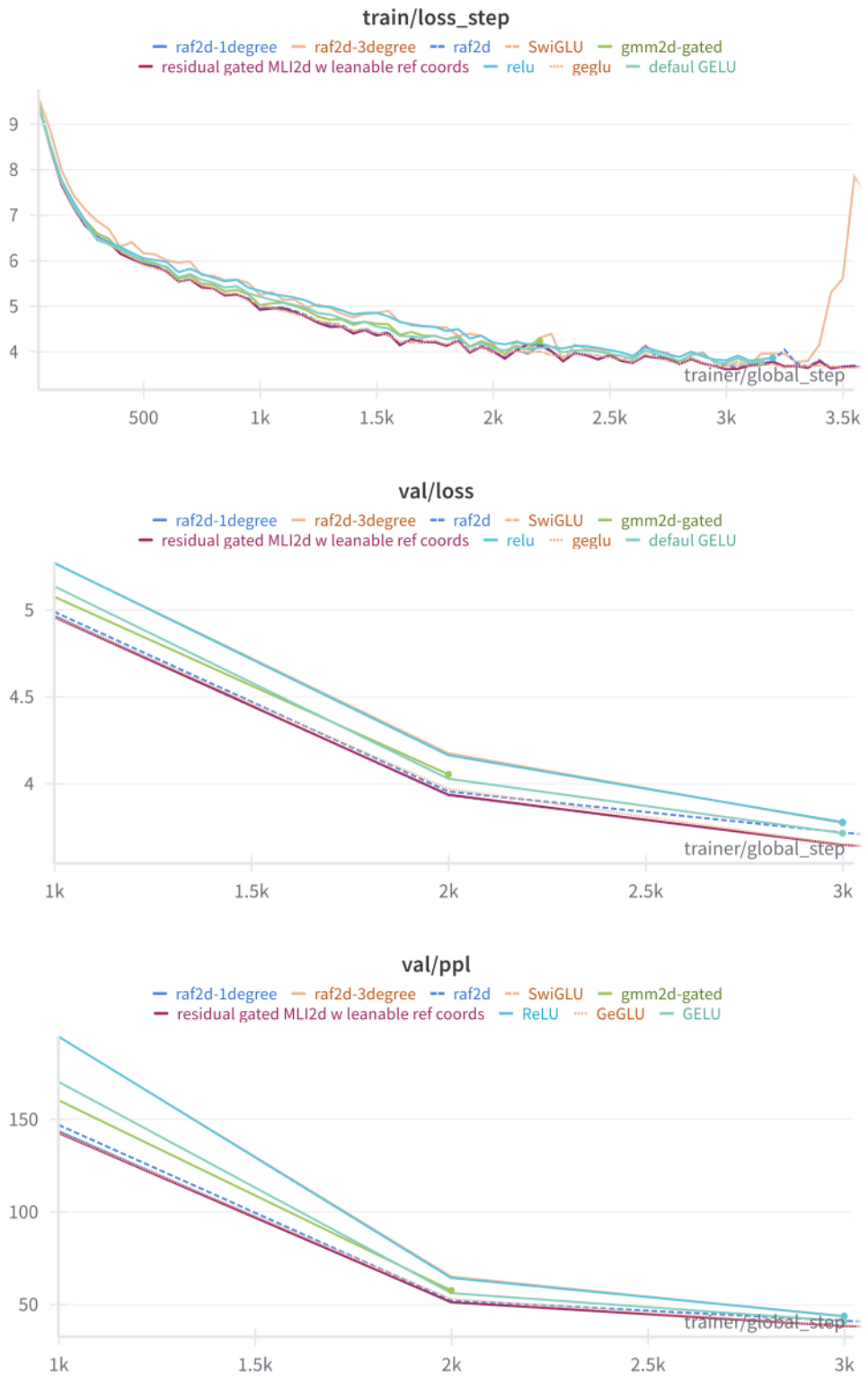


Рис. 4.1: Динаміка навчання моделі Transformer з різними функціями активації. (Зверху) Значення loss-функції на тренуванні. (Посередині) Значення loss-функції на валідації. (Знизу) Валідаційна перплексія.

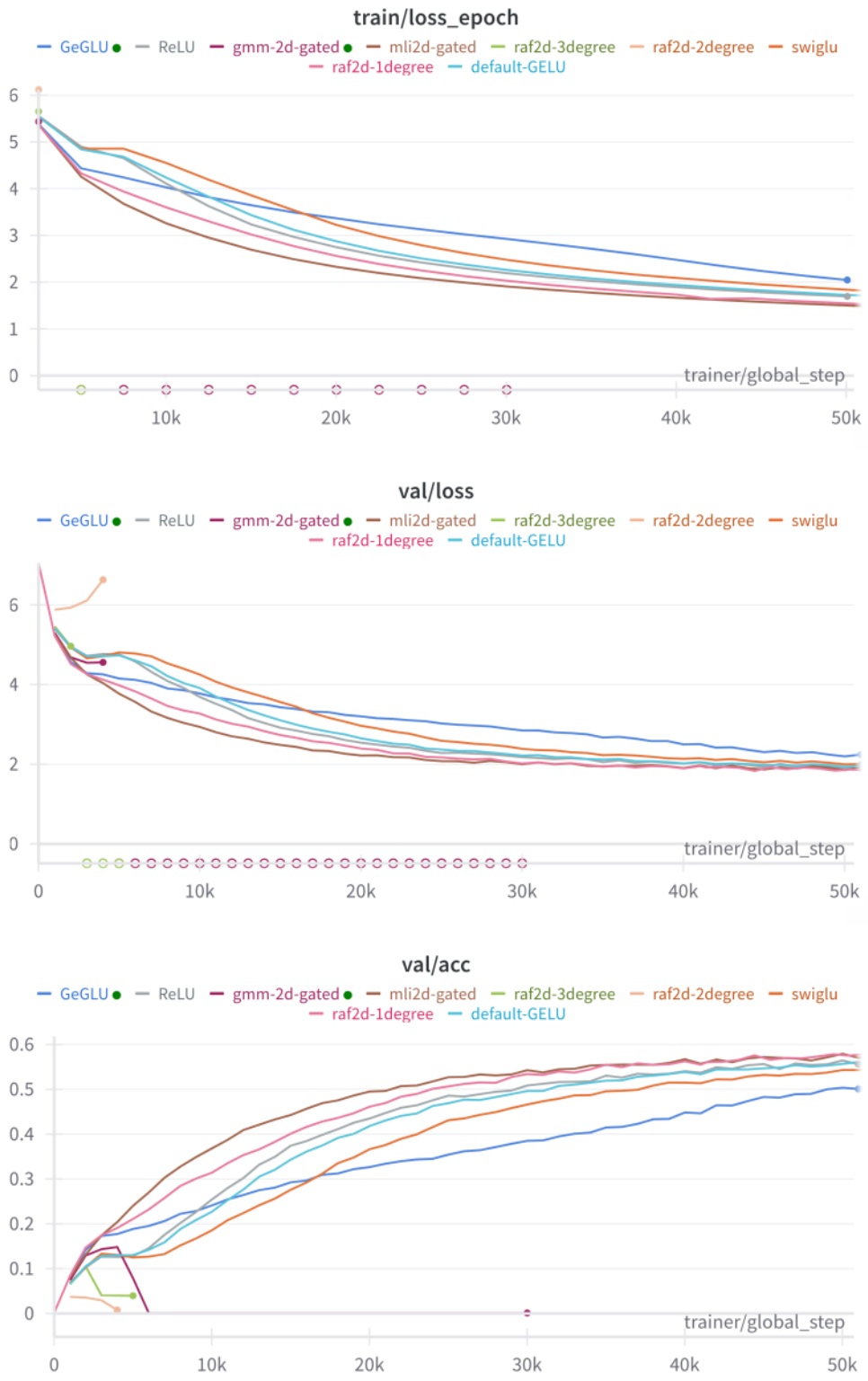


Рис. 4.2: Динаміка навчання моделі Vision Transformer з різними функціями активації. (зверху) Втрати на тренуванні. (посередині) Валідаційні втрати. (знизу) Точність валідації.

РОЗДІЛ 5
АДВЕРСАРІАЛЬНА СТІЙКІСТЬ ЗАПРОПОНАВАНИХ ПАРАМЕТРИЗАЦІЙ

5.1. Огляд адверсаріальної стійкості

5.1.1. Визначення та математичні основи адверсаріальної стійкості

Адверсаріальна стійкість (англ. adversarial robustness) є властивістю машинного навчання, яка дозволяє моделям залишатися стабільними та ефективними навіть у присутності адверсаріальних атак. Адверсаріальні атаки є техніками, які змінюють вхідні дані таким чином, щоб викликати помилкові рішення моделі, зберігаючи при цьому ці зміни непомітними для людини.

Формально, нехай $f : \mathbb{R}^d \to \mathbb{R}^k$ є класифікатором, що приймає вхідні дані $x \in \mathbb{R}^d$ і видає ймовірнісний розподіл класів $\hat{y} = f(x)$. Адверсаріальна атака на x полягає у знаходженні невеликого збурення δ такого, що модель дає помилковий результат: $f(x+\delta) \neq f(x)$, при цьому значення $\|\delta\|$ є малим.

5.1.2. Математична модель адверсаріальної стійкості

Адверсаріальна стійкість може бути оцінена через оптимізаційну проблему:

$$\delta^* = \arg\max_{\|\delta\| \leqslant \epsilon} L(f(x+\delta), y)$$

де L - функція втрат (наприклад, крос-ентропія), y - справжня мітка класу, а ϵ - допустимий радіус збурення.

Модель вважається адверсаріально стійкою, якщо максимальне значення функції втрат від збурень δ^* є невеликим. Іншими словами, стійкість моделі

можна оцінити через мінімізацію впливу адверсаріальних збурень:

$$\min_f \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\delta\|\leqslant\epsilon} L(f(x+\delta), y) \right]$$

де $\mathcal{D}$ - розподіл даних.

5.1.3. Підходи до підвищення адверсаріальної стійкості

Для підвищення адверсаріальної стійкості існує декілька загальновідомих методів:

- **Адверсаріальне навчання**: включення адверсаріальних прикладів у тренувальний набір даних [11].
- **Регуляризація**: використання регуляризаційних технік, таких як ℓ_2-норма або ентропійна регуляризація [4].
- **Рандомізоване згладжування**: створення гладких моделей шляхом додавання гауссового шуму до вхідних даних [124].

Адверсаріальна стійкість має велике значення для безпеки та стабільності систем машинного навчання, особливо у критичних додатках, таких як розпізнавання облич, автономні транспортні засоби та кібербезпека.

5.1.4. Алгоритм AutoAttack

Алгоритм AutoAttack [9] — це набір різноманітних атакуючих алгоритмів для надійного оцінювання адверсаріальної стійкості моделей машинного навчання, що об'єднує декілька різноманітних атак без параметрів. Він був розроблений для того, щоб забезпечити автоматичне та точне оцінювання без необхідності налаштування гіперпараметрів для кожного нового захисту. AutoAttack складається з чотирьох компонентів: дві нові версії APGD (APGD-CE та APGD-DLR), white-box атака FAB і black-box атака Square Attack. Кожна з цих атак має різні підходи та характеристики, що дозволяє досягти надійного і різноманітного тестування адверсаріальної стійкості моделей.

Звязок з PGD

Розглянемо класифікатор $g : D \subseteq \mathbb{R}^d \to \mathbb{R}^K$, який приймає рішення за правилом $\arg\max_{k=1,\dots,K} g_k(\cdot)$, та точку $x_{\text{orig}} \in \mathbb{R}^d$, яка правильно класифікується як клас c. Враховуючи метрику $d(\cdot,\cdot)$ та $\epsilon > 0$, модель загрози (допустима множина атак) визначається як $\{z \in D | d(x_{\text{orig}}, z) \leqslant \epsilon\}$. Тоді z є адверсаріальним прикладом для g в точці x_{orig} щодо моделі загрози, якщо:

$$\arg\max_{k=1,\dots,K} g_k(z) \neq c, \quad d(x_{\text{orig}}, z) \leqslant \epsilon \quad \text{та} \quad z \in D. \tag{5.1.1}$$

Для знаходження z зазвичай визначають функцію-замінник L так, що розв'язання задачі оптимізації з обмеженнями:

$$\max_{z \in D} L(g(z), c) \quad \text{що задовольняє} \quad d(x_{\text{orig}}, z) \leqslant \epsilon, \quad z \in D \tag{5.1.2}$$

призводить до того, що z не буде класифікований як клас c. У задачах класифікації зображень найпопулярніші моделі загрози базуються на l_p-відстанях, тобто $d(x, z) := \|z - x\|_p$, а $D = [0, 1]^d$.

Оскільки проекція на l_p-кулю для $p \in \{2, \infty\}$ доступна в закритій формі, задачу (5.1.2) можна розв'язати за допомогою проекційного градієнтного спуску (PGD).

Нехай $f : \mathbb{R}^d \longrightarrow R$, $\mathcal{S} \subset R^d$ та задачу $\max\limits_{x \in \mathcal{S}} f(x)$, ітерації PGD визначаються для $k = 1, \dots, N_{\text{iter}}$ як $x^{(k+1)} = P_{\mathcal{S}}\left(x^{(k)} + \eta^{(k)} \nabla f(x^{(k)})\right)$, де $\eta^{(k)}$ - це розмір кроку на ітерації k, а $P_{\mathcal{S}}$ - проєкція на $\mathcal{S}$. Використовуючи кросентропійну (CE) функцію втрат як цільову функцію L, [6, 11] ввели PGD-атаку. У їхньому формулюванні $\eta^{(k)} = \eta$ для кожного k, тобто розмір кроку є фіксованим, а як початкова точка x_0 використовується або x_{orig}, або $x_{orig} + \zeta$, де ζ випадково вибирається так, щоб x_0 задовольняло обмеження. Крім того, градієнтний спуск виконується відповідно до норми моделі загрози (наприклад, для l_∞ використовується знак градієнта) [9].

Альтернативні функції втрат Якщо x має правильний клас y, то крос-ентропійна функція втрат для x має вигляд:

$$\mathrm{CE}(x, y) = -\log p_y = -z_y + \log\big(\sum_{j=1}^{K} e^{z_j}\big), \tag{5.1.3}$$

де $p_i = e^{z_i} / \sum_{j=1}^{K} e^{z_j},\ i = 1, \ldots, K$, що є інваріантним до зсувів логітів z, але не до масштабування, подібно до градієнта відносно x, який дорівнює

$$\nabla_x \mathrm{CE}(x, y) = (-1 + p_y)\, \nabla_x z_y + \sum_{i \neq y} p_i \nabla_x z_i. \tag{5.1.4}$$

Якщо $p_y \approx 1$ і, відповідно, $p_i \approx 0$ для $i \neq y$, тоді $\nabla_x \mathrm{CE}(x, y) \approx \mathbf{0}$ (явище затухання градієнта було описане в роботі [12]). Важливим також є те, що можна досягти $p_y \approx 1$ з класифікатором $h = \alpha g$, еквівалентним g (тобто вони приймають однакове рішення для кожного x), але масштабованим параметром $\alpha > 0$. В своїй роботі [9] показують, що феномен описаний вище може призвести до переоцінки стійкості.

Функцію втрат CW [12] визначено як

$$CW(x, y) = -z_y + \max_{i \neq y} z_i. \tag{5.1.5}$$

яка на відміну від втрати CE, має пряму інтерпретацію з точки зору рішення класифікатора. Якщо існує адверсаріальний приклад, тоді глобальний максимум втрати CW є позитивним. Однак втрата CW не є масштабно-інваріантною, і тому екстремальне масштабування знову може бути використано для маскування градієнта.

Була запропонована функція втрат зі співвідношенням логітів (англ. Difference of Logits Ratio Loss або DLR) [9], яка є як інваріантною до зсувів і масштабувань, і таким чином має ті самі ступені свободи, що й рішення класифікатора:

$$\mathrm{DLR}(x, y) = -\frac{z_y - \max\limits_{i \neq y} z_i}{z_{\pi_1} - z_{\pi_3}}, \tag{5.1.6}$$

де π є порядком компонент z у спадному порядку. Потрібну інваріантність до зсувів у цій функції втрат досягається за рахунок різниці логітів також у знаменнику. Максимізація DLR відносно x дозволяє знайти точку, класифіковану не як клас y (DLR є позитивним лише якщо $\text{argmax}_i z_i \neq y$) і, як тільки це досягається, мінімізує оцінку класу y порівняно з іншими класами. Якщо x правильно класифіковано, ми маємо $\pi_1 \equiv y$, так що $\text{DLR}(x, y) = -\frac{z_y - z_{\pi_2}}{z_y - z_{\pi_3}}$ і $\text{DLR}(x, y) \in [-1, 0]$. Роль нормалізації $z_{\pi_1} - z_{\pi_3}$ полягає в тому, щоб підштовхнути z_{π_2} до $z_y = z_{\pi_1}$, оскільки він віддає перевагу точкам, для яких $z_y \approx z_{\pi_2} > z_{\pi_3}$, і тому має схильність до зміни рішення. Крім того, в роботі [9] адаптують функцію втрат DLR для помилкової класифікації в цільовий клас t за допомогою

$$\text{Targeted-DLR}(x, y) = -\frac{z_y - z_t}{z_{\pi_1} - (z_{\pi_3} + z_{\pi_4})/2}. \tag{5.1.7}$$

Таким чином, зберігається як зсувна, так і масштабна інваріантність функції-втрат DLR, одночасно прагнучи отримати $z_t > z_y$, і модифікуючи знаменник у (5.1.6), щоб забезпечити не константність функції-втрат.

5.1.4.1. Компоненти AutoAttack

AutoAttack [9] складається з:

- APGDCE без випадкових перезапусків,
- APGDT на основі функції втрат DLR з 9 цільовими класами,
- FABT з 9 цільовими класами,
- Square Attack з 5000 запитами.

FABT: Цільова версія FAB (FABT), описана в 1.1.1 розділі, розглядає лише лінеаризацію гіперплощини між цільовим та правильним класами, що знижує обчислювальну складність і вимоги до пам'яті.

Square Attack: Square Attack — атака, описана в підрозділі 1.1.2 , яка використовує випадковий пошук і не потребує жодної апроксимації градієнта.

Вона перевершує інші атаки чорного ящика за ефективністю запитів і успішністю та показала конкурентоспроможність із білими атаками.

5.2. Метрики та бенчмарки для оцінки адверсаріальної стійкості

Одним з існуючих бенчмарків для оцінки адверсаріальної стійкості є **RobustBench** [125]. Це стандартизований бенчмарк, який базується на використанні AutoAttack, ансамблю white-box та black-box атак, що було показано ефективнішим у порівнянні з іншими методами оцінки стійкості.

5.2.1. Математичні формули та визначення

Нехай $x \in \mathbb{R}^d$ — вхідний вектор, а $y \in \{1, \ldots, C\}$ — його правильна мітка. Для класифікатора $f : \mathbb{R}^d \to \mathbb{R}^C$, успішне адверсаріальне збурення відносно множини збурень $\Delta \subseteq \mathbb{R}^d$ визначається як вектор $\delta \in \mathbb{R}^d$, що задовольняє умови:

$$\arg \max_{c \in \{1,\ldots,C\}} f(x+\delta)_c \neq y \quad \text{та} \quad \delta \in \Delta.$$

Стійкість моделі вимірюється через точність, яка визначається як частка даних, на яких класифікатор f передбачає правильний клас для всіх можливих збурень з множини Δ.

5.2.2. Модель загроз

Модель загроз визначається множиною допустимих збурень Δ. Найбільш поширеними є ℓ_p-нормовані збурення, тобто $\Delta_p = \{\delta \in \mathbb{R}^d \mid \|\delta\|_p \leqslant \epsilon\}$. Важливою метою є забезпечення того, щоб справжня мітка залишалася незмінною для всіх точок $x+\delta$ в межах множини збурень. Стійкість щодо невеликих ℓ_p-збурень є необхідною, але недостатньою умовою стійкості, яка критикується в літературі.

5.2.3. Оцінка захистів

Для оцінки стійкості моделей використовується AutoAttack, який є ансамблем чотирьох атак, що послідовно виконуються:

1. Модифікація PGD атаки з автоматичним налаштуванням кроку.
2. Використання функції втрат DLR (Difference of Logits Ratio).
3. Цілеспрямована версія FAB атаки.
4. Black-box атака Square Attack.

Кожна наступна атака виконується на точках, для яких не було знайдено адверсаріальних прикладів попередніми атаками.

5.2.4. Обмеження

Для забезпечення стандартизованої оцінки алверсаріальної стійкості, RobustBench накладає певні обмеження на тип моделей, що розглядаються:

- Моделі повинні мати ненульові градієнти відносно входів.
- Моделі повинні бути детермінованими.
- Моделі не повинні містити оптимізаційні цикли у процесі передбачення.

5.3. Адверсаріальна стійкість запропонованих параметризацій

У цьому розділі описуються налаштування експериментів для оцінки запропонованих функцій активації та блока уваги Mega (Moving Average Equipped Gated Attention). Експерименти проводилися на датасеті ImageNet, використовуючи різні параметризації для базової моделі Vision Transformer.

Тренувались наступні моделі:

- Базова модель Vision Transformer.
- Vision Transformer з функцією активації MLI
- Vision Transformer з блоком уваги Mega.

- Vision Transformer з блоком уваги Mega та функцією активації MLI

Для всіх експериментів використовувалася архітектура Vision Transformer з прихованим розміром 768, 12 шарами, 12 головами уваги та 16x16 патчами зображень.

Для оцінки продуктивності моделей використовувалися наступні метрики:

- Стандартна точність.
- Адверсаріальна точність після проведення бенчмарків з використанням RobustBench, що включає:
  - Адверсеріальна точність під загрозою l_2.
  - Адверсеріальна точність під загрозою l_∞.

Показники адверсеріальної точності оцінювалися за різними бюджетами пертурбацій.

5.3.1. Процедура навчання

Усі моделі тренувалися протягом 19 епох, що відповідає приблизно 50 тисяч ітерацій оптимізатора Адам. Для навчання використовувався набір даних ImageNet, зображення якого були масштабовані до розміру 256x256. Кожне зображення оброблялося у вигляді сітки з 16x16 патчів, кожен з яких проходив лінійну трансформацію та перетворювався у 1-вимірну послідовність перед передачею до моделі трансформера.

5.3.2. Результати експериментів

У цьому підрозділі представлені результати експериментів для оцінки адверсеріальної стійкості та стандартної точності різних моделей Vision Transformer (ViT), включаючи базову модель, модель з блоком уваги MEGA, та моделі з багатовимірною функцією активації $MLI2d$.

На рис. 5.1 представлені графіки адверсеріальної топ-1 точності для рі-

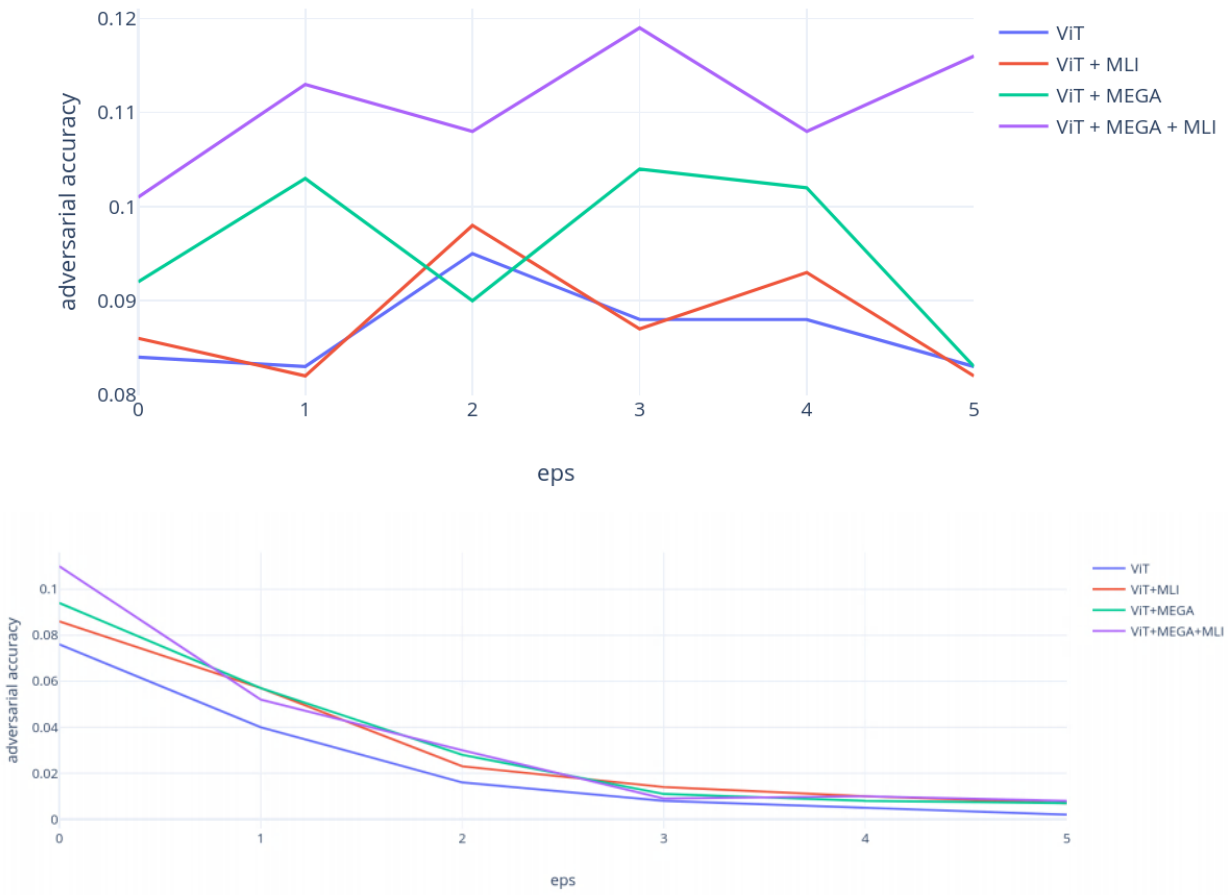


Рис. 5.1: Адверсеріальна точність під загрозою l_2 (вгорі) та l_∞ (внизу) для різних моделей при різних бюджетах пертурбацій (ϵ).

зних моделей при загрозах l_2 та l_∞ відповідно, з різними бюджетами пертурбацій (ϵ). Кожен графік відображає 4 моделі: базовий ViT, ViT+MEGA, ViT+інтерполяційна функція активації, ViT+MEGA+інтерполяційна функція активації.

Далі у таблиці 5.1 представлена стандартна точність для всіх розглянутих моделей.

Результати показують, що додавання блоку уваги MEGA та використання інтерполяційних функцій активації покращують класифікаційну точність, зберігаючи адверсеріальну стійкість моделей при різних загрозах та бюджетах пертурбацій на такому ж або вищому рівні ніж базова модель ViT.

Таблиця 5.1

Стандартна точність для різних моделей Vision Transformer.

Модель	**точність**
Базовий ViT	55.67%
ViT+MEGA	59.0‘%
ViT+інтерполяційна функція активації	57.93%
ViT+MEGA+інтерполяційна функція активації	**61.02%**

5.4. Адверсаріальне тренування

Адверсаріальне навчання є ключовою і найбільш поширеною технікою для підвищення адверсаріальної стійкості моделей. Воно включає тренування моделей на адверсаріальних прикладах, що дозволяє їм виявляти та правильно обробляти збурені дані.

5.4.1. Основні концепції і походження адверсаріального навчання

Основна ідея адверсаріального навчання полягає у тому, щоб під час тренування моделі включати збурені приклади $x' = x + \delta$, де δ є адверсаріальним збуренням, яке знаходиться у межах допустимого радіусу(бюджету пертурбацій) ϵ. Таким чином, мета адверсаріального навчання полягає у мінімізації loss-функції на адверсаріальних прикладах.

Наразі адверсаріальне навчання широко прийнято як найбільш ефективний метод на практиці для покращення адверсаріальної стійкості моделей глибокого навчання [126].

Початкову ідею адверсаріального навчання вперше було висвітлено у [3], де нейронні мережі навчаються на суміші адверсаріальних та чистих прикладів. [4] пішли далі і запропонували FGSM для створення адверсаріальних прикладів під час навчання. Проте, їхні навчені моделі залишаються вразливими до ітеративних атак [127], оскільки ці підходи використовували лінійну

функцію для апроксимації loss-функції, що призводить до різкого викривлення поблизу точок даних на поверхні рішень відповідних глибоких моделей. Наявність різкого викривлення також відома як градієнтне маскування [36].

На відміну від робіт, де моделі навчалися на суміші чистих і адверсаріальних даних, є низка досліджень де досліджують моделі, які навчаються лише з адверсаріальними даними. Вперше, [128] визначили задачу мін-макс, де процес навчання примушений мінімізувати помилку класифікації проти супротивника, який змінює вхідні дані і максимізує помилку класифікації. Вони також зазначили, що ключ до розв'язання цієї задачі мін-макс полягає у знаходженні сильних адверсаріальних прикладів. [129] розглянули цю задачу мін-макс з точки зору надійної оптимізації і запропонували структуру адверсаріального навчання. Формулювання показано нижче:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in B(x,\varepsilon)} \mathcal{L}_{ce}(\theta, x+\delta, y) \right], \tag{5.4.1}$$

де $(x, y) \sim \mathcal{D}$ представляє навчальні дані, вибрані з розподілу $\mathcal{D}$, а $B(x, \varepsilon)$ - це допустимий набір збурень, виражений як $B(x, \varepsilon) := \{x + \delta \in \mathbb{R}^m \mid \|\delta\|_p \leqslant \varepsilon\}$. [11] дали розумне тлумачення цього формулювання: внутрішня задача максимізації полягає у знаходженні найгірших зразків для даної моделі, а зовнішня задача мінімізації полягає у навчанні моделі, стійкої до адверсаріальних прикладів [130].

З цією метою [11] використали проектовану атаку на основі градієнта, відому як атака PGD, для розв'язання внутрішньої проблеми наступним чином:

$$x^{t+1} = \text{Proj}_{x+B(x,\varepsilon)} \left(x^t + \alpha \,\text{sign} \left(\nabla_{x^t} \mathcal{L}_{ce} \left(\theta, x^t, y \right) \right) \right), \tag{5.4.2}$$

де t - поточний крок, а α - розмір кроку. Далі вони дослідили внутрішню задачу максимізації з точки зору ландшафту адверсаріальних прикладів та надали як теоретичні, так і емпіричні докази придатності локальних максимумів з PGD. Через великі експерименти їхній підхід (PGD-AT) значно підвищив адверсаріальну стійкість моделей глибокого навчання проти широкого спектра атак, що є важливою віхою у методах адверсаріального навчання.

Більшість похідних робіт наслідували їхні розробки та налаштування, тому PGD-AT став критичним еталоном і вважається стандартним методом адверсаріального навчання на практиці [130].

5.4.2. Таксономія адверсаріального навчання

Адверсаріальна регуляризація Ідея адверсаріальної регуляризації вперше з'являється у [4]. Окрім loss-функції крос-ентропії, вони додали регуляризаційний термін у цільову функцію, який базується на FGSM і виражається як $\mathcal{L}\left(\boldsymbol{\theta}, x+\epsilon\,\mathrm{sign}\left(\nabla_x \mathcal{L}(\boldsymbol{\theta}, x, y)\right)\right.$. [131] розширили цей регуляризаційний термін, заснований на FGSM, контролюючи частку адверсаріальних прикладів у тренувальних батчах, що дозволило тренувати моделі на датасеті ImageNet. Ефективність їхніх методів була підтверджена на одноетапних атаках, оскільки автори вважали, що лінійність нейронних мереж сприяє існуванню адверсаріальних прикладів [4]. Однак [132] обчислили абсолютну різницю між адверсаріальною втратою та її першопорядковим розкладом Тейлора, зробивши висновок, що більш стійкі моделі зазвичай мають менші значення локальної лінійності. Відповідно, вони замінили регуляризацію на основі FGSM на регуляризацію локальної лінійності для адверсаріальної стійкості [130].

На відміну від попередніх методів, [1] декомпозували помилку моделі $\mathcal{R}_{\text{rob}}$ як суму звичайної класифікаційної помилки $\mathcal{R}_{\text{nat}}$ та помилки на межі $\mathcal{R}_{\text{db}}$. Помилка на межі виникає, коли відстань між даними та межею рішень є досить малою (меншою за ϵ), що є також причиною існування адверсаріальних прикладів. Тому вони сформулювали оптимізаційну задачу на основі компромісу, мінімізуючи сурогатну функцію втрат (англ. Tradeoff-inspired Adversarial Defense via Surrogate-loss minimization, або TRADES) $\mathcal{R}_{\text{db}}$, розв'язуючи наступну задачу:

$$\min_f \mathbb{E}\{\mathcal{L}(f(x), y) + \max_{x' \in \mathbb{B}(x,\epsilon)} \mathcal{L}\left(f(x), f\left(x'\right)\right)/\lambda\}, \tag{5.4.3}$$

де λ - це коефіцієнт, який визначає силу регуляризації. Ця декомпозиція виявилася ефективною, і TRADES перевершує PGD-AT на CIFAR-10 з зменшенням помилок на 10%. Однією з проблем TRADES є те, що регуляризаційний термін призначений для зближення натуральних прикладів та їх адверсаріальних аналогів, незалежно від того, чи правильно класифіковані натуральні дані. [133] досліджували вплив неправильно класифікованих прикладів і запропонували тренувальний метод MART, який акцентує увагу на неправильно класифікованих прикладах з вагами $1 - \mathcal{P}_y(x, \theta)$, де $\mathcal{P}_y(x, \theta)$ - це ймовірність правильного класу y [130].

Через посилення глибоких моделей, непомітні шуми можуть призвести до значних змін у просторі ознак [4]. Деякі роботи аналізують адверсаріальне навчання з точки зору представлення. [134] запропонували метод Adversarial Logit Pairing (ALP), який сприяє схожості логітів для пар прикладів. Проте ALP спочатку не був корисним через неправильне формулювання цілей адверсаріального навчання [135]. Для подальшого покращення представлень натуральних даних та їх адверсаріальних аналогів, [136] прийняли популярну тріплетну функцію втрат для регуляризації, яка використовує адверсаріальні приклади як опорні точки [130]. Адверсаріальна регуляризація є важливим варіантом адверсаріального навчання [129]. Порівняно з оригінальним формулюванням адверсаріального навчання, адверсаріальна регуляризація є більш гнучкою і вимагає глибокого розуміння адверсаріальної стійкості. Крім того, декомпозиція помилки дійсно прокладає шлях для невірно класифікованих оригінальних даних для покращення адверсаріальної стійкості [130].

Адверсаріальне навчання на основі курикулуму. Згідно з формулюванням адверсаріального навчання, внутрішня задача завжди намагається знайти зразки в найгіршому випадку. [137] підняли питання, чи завжди ці зразки найгіршого випадку підходять для адверсаріального навчання? Автори виявили, що адверсаріальні приклади, створені сильними атаками, значно перетинають межу рішення і є близькими до натуральних даних. Оскільки PGD-

AT використовує лише адверсаріальні приклади для навчання, це призводить до перенавчання на адверсаріальних прикладах [138]. Для пом'якшення перенавчання дослідники адаптували ідею курикулумного навчання до адверсаріального навчання. [138] запропонували Curriculum Adversarial Training (CAT), припускаючи, що PGD з більшою кількістю кроків створює сильніші адверсаріальні приклади. Починаючи з невеликої кількості кроків, CAT поступово збільшує кількість ітерацій PGD, доки модель не досягне високої точності проти поточної атаки. На відміну від CAT, метод Friendly Adversarial Training (FAT) [137] використовує ранню зупинку під час виконання атак PGD і повертає адверсаріальні дані поблизу межі рішення для навчання. Обидва методи, CAT і FAT, коригують силу атак практичним способом, де кількісного критерію не вистачає. З точки зору збіжності, [139] розробили First-Order Stationary Condition (FOSC) для оцінки якості збіжності внутрішньої задачі максимізації. Чим ближче FOSC до 0, тим сильніша атака [130].

Такі методи на основі курикулуму допомагають покращити генералізацію чистих даних, зберігаючи при цьому адверсаріальну стійкість. Одна з можливих причин їхнього успіху полягає в тому, що слабкі атаки на ранніх стадіях навчання пов'язані з генералізацією [139]. Крім того, що знижують перенавчання, методи на основі курикулуму скорочують час навчання завдяки змінній кількості ітерацій PGD для розв'язання внутрішньої задачі максимізації [130].

Адверсаріальне навчання ансамблем [127] вперше ввели ансамблеве навчання в адверсаріальне навчання, назване Ensemble Adversarial Training (EAT), де навчальні дані доповнюються адверсаріальними прикладами, створеними з різних цільових моделей замість однієї моделі. Перевага EAT полягає в тому, що він допомагає пом'якшити різкі викривлення, викликані одноетапними атаками, такими як FGSM. Однак взаємодія між різними цільовими моделями не береться до уваги [127]. Зокрема, стандартно навчені цільові моделі можуть мати схожі передбачення або представлення [140] та

ділити адверсаріальний підпростір [141], що потенційно погіршує ефективність EAT. В основі такі ансамблеві методи корисні для наближення оптимального значення внутрішньої задачі максимізації в адверсаріальному навчанні. Як доведено в [127], моделі, навчені за допомогою EAT, мають кращі здатності до генералізації незалежно від типів збурень. На завершення, додавання кількості та різноманітності цільових моделей у навчанні є практичним і корисним способом наближення простору адверсаріальних прикладів, який складно описати явно [130].

Адверсаріальне навчання з адаптивним ϵ. Як показано в рівнянні (5.4.1), параметри атак визначені заздалегідь і фіксовані під час навчання, наприклад ϵ. Деякі роботи [142, 143] стверджують, що деякі вхідні дані можуть мати різну внутрішню стійкість, тобто різні відстані до межі рішень класифікатора; однак, адверсаріальне навчання з фіксованим ϵ обробляє всі дані однаково [130]. З огляду на індивідуальні характеристики адверсаріальної стійкості, дослідники пропонують виконувати адверсаріальне навчання на рівні прикладів. [142] вперше представили Instance Adaptive Adversarial Training (IAAT), де ϵ вибирається максимально можливим, забезпечуючи, що зображення в межах ϵ-околу x належать до одного класу. Ця стратегія допомагає IAAT пом'якшити компроміс між робастністю та точністю, хоча є невелике зниження робастності. На відміну від IAAT, інша робота під назвою Margin Maximization Adversarial Training (MMA) [143] безпосередньо максимізує відстані між точками даних і межею рішень моделі, що оцінюється адверсаріальними збуреннями з найменшими величинами. Спосіб вибору ϵ в MMA є більш розумним, оскільки ϵ достатньо малий, і така мала ϵ у просторі навряд чи суттєво змінить класи зображень, особливо для високоякісних зображень. Наступна робота, Customized Adversarial Training (CAT) [144] далі застосовує адаптивну невизначеність міток для запобігання надмірно впевненим прогнозам на основі адаптивного ϵ [130].

Адверсаріальне навчання з адаптивним ϵ є хорошою розробкою. Однак

емпіричні дані показують, що багато стандартних наборів даних розподільчо розділені, тобто відстані між класами більші, ніж ϵ, що використовується для атак [145]. Це відображає обмеження поточних методів адверсаріального навчання у знаходженні відповідних меж рішень [130].

Адверсаріальне навчання на немаркованих даних Одне з ключових спостережень у методах контрольованого адверсаріального навчання [1, 11] полягає в тому, що адверсаріальна точність під час тестування значно нижча, ніж під час навчання. Існує велика різниця в генералізації в адверсаріальному навчанні. Робота [146] досліджувала цю проблему з точки зору складності вибірки. Теоретично доведено, що адверсаріально стійке навчання вимагає значно більших даних, ніж стандартне навчання. Однак, якісні набори даних з мітками дорого збирати. Як наслідок, з'явилися кілька робіт, які досліджують можливість навчання з додатковими немаркованими даними [130].

Слідом за аналізом моделей Гаусса в [146], кілька робіт [147–149] теоретично показують, що немарковані дані значно зменшують розрив у складності вибірки між стандартним і адверсаріальним навчанням. Вони поділяють ту ж ідею декомпозиції адверсаріальної стійкості, як TRADES, і використовують немарковані дані для стабільності, тоді як марковані дані для класифікації. Емпірично вони досліджували вплив різних факторів на адверсаріальне навчання, таких як шум міток, зсув розподілу та кількість додаткових даних [130]. Однак теоретичні чи емпіричні гарантії того, скільки саме додаткових даних потрібно, все ще відсутні. Крім того, не варто забувати про вартість таких методів, включаючи збір даних та адверсаріальне навчання на наборах даних, що у кілька разів більші за оригінальні [130].

5.4.3. Рандомізоване згладжування

Рандомізоване згладжування є підходом, що використовує додавання випадкового шуму до вхідних даних під час навчання. Цей метод дозволяє моделі стати більш стійкою до збурень, зберігаючи при цьому високу точність

[124]. Формально, цей метод можна описати як:

$$f(x) = \mathbb{E}_{\eta \sim \mathcal{N}(0,\sigma^2 I)}[g(x+\eta)]$$

де g - базовий класифікатор, η - гауссовий шум з дисперсією σ^2.

5.4.4. Метод Mixup

Підхід Mixup [150] — це метод аугментації даних, що спрямований на покращення генералізації моделей машинного навчання. Основна ідея Mixup полягає у створенні нових тренувальних прикладів шляхом лінійної інтерполяції між парами існуючих прикладів та їх відповідними мітками.

Нехай (x_i, y_i) та (x_j, y_j) — дві випадково обрані пари вхідних даних і міток з тренувального набору. Тоді новий вхідний вектор $\tilde{x}$ та відповідна мітка $\tilde{y}$ визначаються як:

$$\tilde{x} = \lambda x_i + (1-\lambda) x_j, \tag{5.4.4}$$

$$\tilde{y} = \lambda y_i + (1-\lambda) y_j, \tag{5.4.5}$$

де $\lambda \in [0,1]$ — випадкова змінна, взята з бета-розподілу $Beta(\alpha, \alpha)$ для деякого $\alpha > 0$.

Метод Mixup підвищує стійкість моделей до шуму та покращує їх здатність до генералізації, зменшуючи можливість перенавчання. Це досягається за рахунок створення нових навчальних прикладів, що забезпечують плавний перехід між класами та сприяють формуванню більш згладжених меж класифікації.

5.4.5. Адверсаріальне міксування

Підхід Adversarial Vertex Mixup (AVmixup) [151] — це метод покращення стійкості моделей машинного навчання до адверсаріальних атак. AVmixup представляє новий підхід до генерації даних із ”м’якими”мітками для тренування моделей, що дозволяє значно підвищити генералізацію. Цей метод

базується на ідеї змішування вхідних векторів у напрямку адверсаріальних прикладів. Даний метод мотивується явищем Adversarial Feature Overfitting (AFO), яке може спричинити погану адверсаріальну узагальнюючу здатність, і автори демонструють, що адверсаріальне тренування може перевищувати оптимальну точку щодо адверсаріальної генералізації, що призводить до AFO.

Використовуючи просту Гауссову модель даних, авторами методу продемонстровано, що необхідно мінімізувати дисперсію представлень ознак для робастної генералізації. Оптимальний параметр моделі з точки зору робастної генералізації відрізняється від параметра моделі, який мінімізує адверсаріальний емпіричний ризик, використовуючи дані, що складаються з робастних та не-робастних ознак. Ми наводимо докази того, що більшість глибоких нейронних мереж не є вільними від AFO.

AVmixup, подібно до Mixup, розширює тренувальний розподіл за допомогою лінійної інтерполяції. На відміну від Mixup, AVmixup визначає віртуальний вектор у напрямку адверсаріального прикладу для кожного вхідного вектора та розширює тренувальний набір даних за допомогою лінійної інтерполяції між віртуальним вектором і вхідним вектором. Цей віртуальний вектор називається адверсаріальною вершиною.

Нехай $\delta_x \in \mathbb{R}^d$ — адверсаріальне збурення для вхідного вектора $x \in \mathbb{R}^d$. Тоді, для масштабу $\gamma \geqslant 1$, адверсаріальна вершина x_{av} визначається як:

$$x_{\text{av}} = x + \gamma \cdot \delta_x. \tag{5.4.6}$$

Після отримання адверсаріальної вершини, AVmixup створює віртуальні тренувальні приклади наступним чином:

Нехай (x, y) — вхідна пара "вектор-мітка". Нехай ϕ — функція згладжування міток. Тоді, для значення α, вибраного з рівномірного розподілу $U(0, 1)$, та параметрів згладжування міток $\lambda_1 \in \mathbb{R}$ та $\lambda_2 \in \mathbb{R}$, віртуальний вхідний

вектор $\hat{x} \in \mathbb{R}^d$ та асоційована мітка $\hat{y} \in \mathbb{R}^k$ будуються за формулами:

$$\hat{x} = \alpha x + (1 - \alpha) x_{\text{av}}, \tag{5.4.7}$$

$$\hat{y} = \alpha \phi(y, \lambda_1) + (1 - \alpha) \phi(y, \lambda_2). \tag{5.4.8}$$

Для функції згладжування міток ϕ ми використовуємо існуючий метод згладжування міток. У випадку k класів алгоритм присвоює $\lambda \in (0, 1)$ істинному класу і рівномірно розподіляє $1 - \lambda/(k - 1)$ на інші класи.

AVmixup покращує адверсаріальну генералізацію моделі за рахунок збільшення різноманітності тренувальних даних. Використання адверсаріальних вершин як додаткових віртуальних точок тренування дозволяє моделі навчитися розрізняти більше ”стійких”ознак, що знижує ризик перенавчання на ”не-стійких”ознаках.

5.5. Ефективність запропонованих параметризацій разом з адверсаріальним тренуванням

У цьому розділі оцінюється ефективність запропонованих параметризацій на основі адверсаріального тренування. Для цього були використані дві моделі: базова ViT та ViT з MEGA-блоком уваги та інтерполяційною функцією $RAF2d_1$. Обидві моделі були натреновані з використанням підходу адверсаріального тренування.

5.5.1. Методи адверсаріального тренування

Для тренування моделей використовувалися наступні техніки адверсаріального тренування:

- випадковий mixup
- згладжена loss-функція крос-ентропії
- адверсаріально змінені зображення за допомогою алгоритму PGD з

бюджетом збурень eps=1/255, 3 ітераціями, та кроком адверсаріального оновлення 1.0. Модель загрози для тренування була L_{inf}.

5.5.2. Архітектура та набір даних

Для експериментів використовувалася та ж архітектура трансформера, що і в попередніх експериментах. Навчальний набір даних був ImageNet.

5.5.3. Результати оцінювання

Для валідації ми наводимо наступні графіки:

- точність (top-1 і top-5)
- Значення loss-функції на тренувальній та валідаційній вибірках
- адверсаріальна точність (top-1 і top-5)
- Значення loss-функції на валідаційній вибірці після адверсаріальної атаки

Крім того, представлені результати RobustBench бенчмарку для обох моделей (таблиця 5.2).

Модель	**точність**	**адв. точність (l_2)**	**адв. точність (l_{inf})**
Baseline ViT	**23.6%**	4.94%	**3.82%**
ViT + MEGA + RAF	20.3%	**5.48%**	3.08%

Таблиця 5.2

Результати RobustBench бенчмарку, для l_2 та l_{inf} нормованих атак, eps=8/255, розмір вибірки 5000 зображень з валідаційного набору даних Imagenet

Графіки 5.2 і 5.3 відображають загальну точність (top-1 і top-5), де видно, що модифікована модель ViT з EMA (*ViT + MEGA + RAF2d (EMA)*) трохи поступається базовим моделям ViT, але не суттєво, а модель без EMA((*ViT + MEGA + RAF2d*) показує нестабільну динаміку тренування, про що свідчить

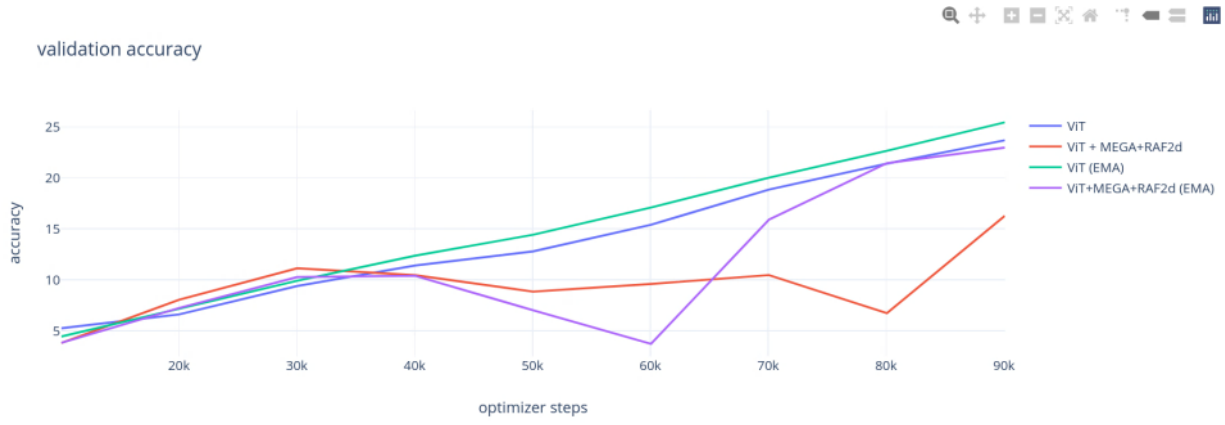


Рис. 5.2: Точність

Рис. 5.3: Top-5 точність

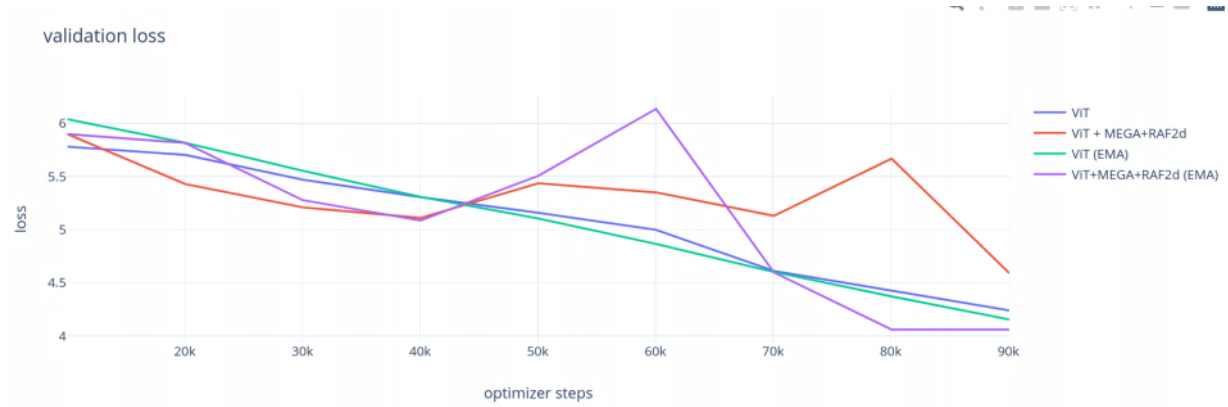


Рис. 5.4: Значення loss-функції

також динаміка функції втрат 5.8. На графіках 5.5 та 5.6 видно, що моделі ViT з розглянутими параметризаціями поступаються в класифікаційні точності базовим моделям. Графік 5.7 теж відображає нестабільність в тренування для $ViT + MEGA + RAF2d$ моделей. Отже, можна зробити наступні спосте-

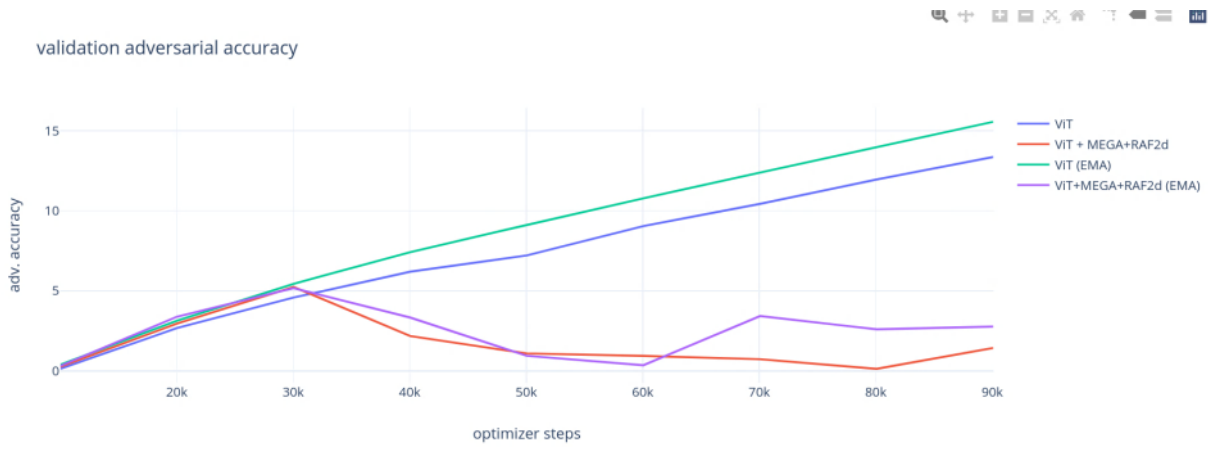


Рис. 5.5: Адверсаріальна Top-1 точність

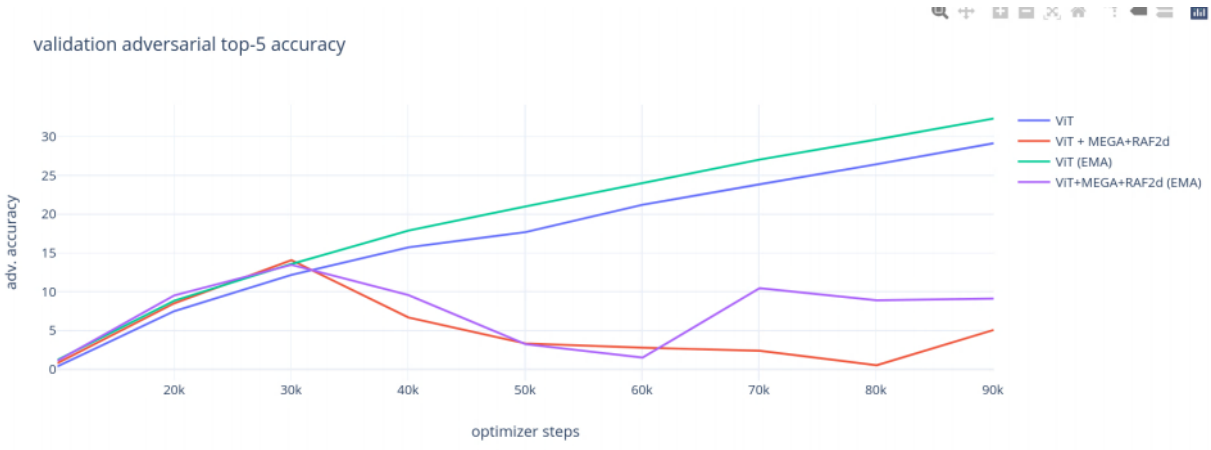


Рис. 5.6: Адверсаріальна Top-5 точність

Рис. 5.7: Адверсаріальні втрати

реження: експоненційне обновлення параметрів покращувало ефективність для обох моделей, по-друге: ViT модель з блоком MEGA та розглянутою багатовимірною функцією активації тренувалась нестабільно, і як наслідок, по-

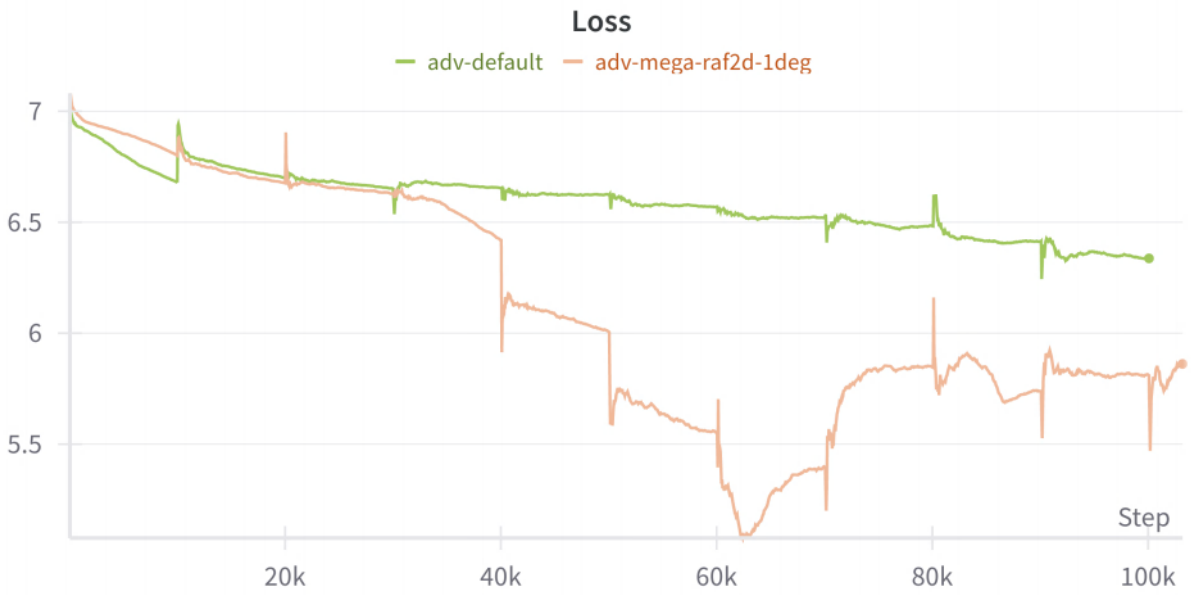


Рис. 5.8: Динаміка loss-функції під час тренування, де *adv-default* це базова ViT модель, а *adv-mega-raf2d-1deg* це ViT з MEGA механізмом уваги та $RAF2d_1$ функцією активацій

казала гіршу класифікаційну і адверсаріальну точність для l_{inf} нормованих атак, незважаючи на кращі показники для звичайного тренування, без адверсаріальних прикладів.

5.6. Адверсаріальне очищення

Адверсаріальне очищення (англ. adversarial purification) - це клас методів захисту, які використовують генеративні моделі для видалення адверсаріальних збурень із зображень. Ці методи не покладаються на конкретну форму атаки або модель класифікації, що дозволяє їм захищати існуючі класифікатори від невідомих загроз. Одним із нових підходів в цій області є використання дифузійних моделей для очищення адверсаріальних прикладів, як запропоновано в роботі [152].

5.6.1. Диффузійні моделі для адверсаріального очищення

Диффузійні моделі складаються з двох процесів: прямого дифузійного процесу, що перетворює дані на шум шляхом поступового додавання шуму до вхідного сигналу, та зворотного генеративного процесу, що починається з шуму і генерує дані шляхом поступового денойзингу.

Нехай $p(x)$ - невідомий розподіл даних, з якого вибирається кожна точка даних $x \in \mathbb{R}^d$. Прямий дифузійний процес $\{x(t)\}_{t\in[0,1]}$ визначається стохастичним диференційним рівнянням (СДР) з додатними приростами часу на фіксованому часовому горизонті $[0, 1]$:

$$dx = f(x, t)dt + g(t)dw, \tag{5.6.1}$$

де початкове значення $x(0) := x \sim p(x)$, $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ - коефіцієнт дрейфу, $g : \mathbb{R} \to \mathbb{R}$ - коефіцієнт дифузії, а $w(t) \in \mathbb{R}^d$ - стандартний процес Вінера.

Розподіл $p_t(x)$ в процесі $x(t)$ з початковим розподілом $p_0(x) := p(x)$. Коефіцієнти $f(x, t)$ та $g(t)$ можна правильно вибрати так, що в кінці дифузійного процесу $x(1)$ буде слідувати стандартному гауссовому розподілу $N(0, I_d)$.

Зворотній генеративний процес визначається зворотним СДР:

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)dw, \tag{5.6.2}$$

де dt - нескінченно малий негативний приріст часу, а $w(t)$ - стандартний зворотний процес Вінера. Вибіркове значення $x(1) \sim N(0, I_d)$ як початкове значення та рішення наведеного СДР від $t = 1$ до $t = 0$ поступово генерують менш шумні дані $x(t)$, поки ми не отримаємо вибірки з розподілу даних, тобто $x(0) \sim p_0(x)$.

Зворотне СДР вимагає знання функції градієнта часу $\nabla_x \log p_t(x)$. Одним із підходів є оцінка $\nabla_x \log p_t(x)$ за допомогою параметризованої нейронної мережі $s_\theta(x, t)$. Відповідно, дифузійні моделі навчаються за допомогою зваженої комбінації денойзингового скорового узгодження (DSM) на різних ча-

сових кроках:

$$\min_{\theta} \int_0^1 \mathbb{E}_{p(x)p_{0t}(x_t|x)} \left[\lambda(t) \| \nabla_{x_t} \log p_{0t}(x_t|x) - s_\theta(x_t, t) \|_2^2 \right] dt,$$

де $\lambda(t)$ - ваговий коефіцієнт, а $p_{0t}(x_t|x)$ - перехідна ймовірність від $x(0) := x$ до $x(t) := x_t$, яка має закриту форму через пряме СДР.

5.6.2. Інференс дифузійної моделі без навчання

Багато семплерів для моделей дифузії покладаються на дискретизацію або SDE зворотного процесу, представленого в 5.6.2. Оскільки вартість вибірки збільшується пропорційно до кількості дискретизованих часових кроків, багато дослідників зосередилися на розробці схем дискретизації, які зменшують кількість часових кроків, одночасно мінімізуючи помилки дискретизації.

Вирішення SDE

Процес генерації DDPM [153, 154] можна розглядати як певну дискретизацію SDE зворотного часу. Прямий процес DDPM дискретизує SDE визначений в 5.6.1, а відповідно зворотній SDE має вигляд

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)(\mathbf{x}_t - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t))\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w} \tag{5.6.3}$$

В роботі [155] було показано, що зворотний ланцюг Маркова, є числовим розв'язувачем SDE.

Noise-Conditional Score Networks (NCSNs) [156] та Critically-Damped Langevin Diffusion (CLD) [157] обидва вирішують SDE зворотного часу, беручи натхнення з динаміки Ланжевена. Зокрема, NCSNs використовують annealed Langevin dynamics (англ. ALD) для ітеративного генерування даних з поступовим зменшенням рівня шуму, доки розподіл згенерованих даних не зійдеться з вихідним розподілом даних. Хоча траєкторії семплювання ALD не є точними розв'язками SDE зворотного часу, вони мають правильні маргінали і, отже, створюють правильні семпли за умови, що динаміка Ланжевена

збігається до свого рівноважного стану на кожному рівні шуму. Метод ALD був покращений Consistent Annealed Sampling (CAS) [158], підходом MCMC на основі оцінки, який краще масштабує часові кроки та доданий шум. Натхненний статистичною механікою, CLD пропонує розширений SDE з допоміжним терміном швидкості, що нагадує динаміку Ланжевена з недозатуханням. Для отримання зворотного часу розширеного SDE, CLD потрібно лише навчити функцію оцінки умовного розподілу швидкості, що є легшим, ніж навчання оцінок даних безпосередньо. Доданий термін швидкості покращує швидкість та якість семплювання.

Метод зворотної дифузії, запропонований у [155], дискретизує SDE зворотного часу так само, як і прямий. Для будь-якої одноетапної дискретизації прямого SDE можна написати загальну форму:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{f}_i(\mathbf{x}_i) + \mathbf{g}_i\mathbf{z}_i, \quad i = 0, 1, \cdots, N-1 \tag{5.6.4}$$

де $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{f}_i$ та $\mathbf{g}_i$ визначаються коефіцієнтами дрейфу/дифузії SDE та схемою дискретизації. Зворотна дифузія пропонує дискретизувати SDE зворотного часу аналогічно прямому SDE, ,

$$\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{f}_{i+1}(\mathbf{x}_{i+1}) + \mathbf{g}_{i+1}\mathbf{g}_{i+1}^t\mathbf{s}_{\theta^*}(\mathbf{x}_{i+1}, t_{i+1}) + \mathbf{g}_{i+1}\mathbf{z}_i \quad i = 0, 1, \cdots, N-1 \tag{5.6.5}$$

де $\mathbf{s}_{\theta^*}(\mathbf{x}_i, t_i)$ - це навчена модель оцінки умовного шуму. Song et al. (2020) [155] доводять, що метод зворотної дифузії є числовим розв'язувачем SDE для зворотного часу в eq:rsde. Цей процес можна застосувати до будь-яких типів прямого SDE, і емпіричні результати показують, що цей семплер працює трохи краще, ніж DDPM [155] для певного типу SDE, званого VP-SDE.

Jolicoeur-Martineau et al. (2021) [159] розробили розв'язувач SDE з адаптивними розмірами кроків для швидшої генерації. Розмір кроку контролюється порівнянням виходу розв'язувача SDE високого порядку з виходом розв'язувача SDE низького порядку. На кожному часовому кроці розв'язувачі

високого та низького порядку генерують нові семпли $\mathbf{x}'_{\text{high}}$ та $\mathbf{x}'_{\text{low}}$ з попереднього семплу $\mathbf{x}'_{prev}$ відповідно. Потім розмір кроку регулюється порівнянням різниці між двома семплами. Якщо $\mathbf{x}'_{\text{high}}$ і $\mathbf{x}'_{\text{low}}$ подібні, алгоритм поверне $\mathbf{x}'_{\text{high}}$ і потім збільшить розмір кроку.

Схожість між $\mathbf{x}'_{\text{high}}$ і $\mathbf{x}'_{\text{low}}$ вимірюється за формулою:

$$E_q = \left\| \frac{\mathbf{x}'_{\text{low}} - \mathbf{x}'_{\text{high}}}{\delta(\mathbf{x}', \mathbf{x}'_{\text{prev}})} \right\|^2 \tag{5.6.6}$$

де $\delta(\mathbf{x}'_{\text{low}}, \mathbf{x}'_{\text{prev}}) \max(\epsilon_{abs}, \epsilon_{rel} \max(|\ \mathbf{x}'_{\text{low}}, |\ \mathbf{x}'_{\text{prev}}|))$, а ϵ_{abs} та ϵ_{rel} є абсолютними та відносними допустимими похибками.

Метод предиктор-коректор, запропонований у [155], вирішує SDE зворотного часу, поєднуючи числові розв'язувачі SDE ("predictor") та ітеративні підходи Марковського ланцюга Монте-Карло (MCMC) ("corrector"). На кожному часовому кроці метод предиктор-коректор спочатку використовує числовий розв'язувач SDE для отримання грубого семплу, а потім використовує "коректор який коригує маргінальний розподіл семплу за допомогою MCMC на основі оцінки. Отримані семпли мають такі ж часові маргінали, як і траєкторії розв'язків SDE зворотного часу, тобто вони еквівалентні в розподілі на всіх часових кроках. Емпіричні результати показують, що додавання коректора на основі методу Ланжевена Монте-Карло є більш ефективним, ніж використання додаткового предиктор без коректорів [155]. Далі вдосконалюють коректор динаміки Ланжевена в [160], де запропонувавши ланжевенівський «збій» етап додавання та видалення шуму, досягли нової найсучаснішої якості семплів на таких даних, як CIFAR-10 і ImageNet-64

Метод adjoint. Розв'язувач ODE може розглядатися як чорний ящик і обчислювати градієнти можна за допомогою методу чутливості спряження [161]. Цей підхід обчислює градієнти, розв'язуючи друге, розширене ODE у зворотному часі, і застосовується до всіх розв'язувачів ODE. Цей підхід масштабується лінійно з розміром проблеми, має низьку вартість пам'яті та явним чином контролює числову похибку.

$$L(z(t_1)) = L\left(z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta)dt\right) = L\left((z(t_0), f, t_0, t_1, \theta)\right) \quad (5.6.7)$$

Для оптимізації L нам потрібні градієнти відносно θ. Перший крок полягає у визначенні того, як градієнт втрат залежить від прихованого стану $z(t)$ у кожний момент часу. Ця величина називається *спряженням* $a(t) = \frac{\partial L}{\partial z(t)}$. Її динаміка задається іншим ODE, яке можна розглядати як миттєвий аналог правила ланцюга:

$$\frac{da(t)}{dt} = -a(t)\frac{\partial f(z(t), t, \theta)}{\partial z} \quad (5.6.8)$$

Ми можемо обчислити $\partial L \partial z(t_0)$ за допомогою іншого виклику розв'язувача ODE. Цей розв'язувач повинен працювати у зворотному напрямку, починаючи з початкового значення $\partial L \partial z(t_1)$. Одним ускладненням є те, що для розв'язування цього ODE потрібно знати значення $z(t)$ на всьому його шляху. Однак, ми можемо просто переобчислити $z(t)$ у зворотному часі разом зі спряженням, починаючи з його кінцевого значення $z(t_1)$.

Обчислення градієнтів відносно параметрів θ вимагає оцінки третього інтегралу, який залежить як від $z(t)$, так і від $a(t)$:

$$\frac{dL}{d\theta} = -\int_{t_1}^{t_0} a(t)^T \frac{\partial f(z(t), t, \theta)}{\partial \theta} dt \quad (5.6.9)$$

Вектор-Якобіан добутки $a(t)^T \frac{\partial f}{\partial z}$ та $a(t)^T \frac{\partial f}{\partial \theta}$ у (5.6.8) та (5.6.9) можуть бути ефективно оцінені за допомогою автоматичного диференціювання, при цьому вартість часу є подібною до оцінки f. Усі інтеграли для розв'язування z, a та $\frac{\partial L}{\partial \theta}$ можуть бути обчислені в одному виклику розв'язувача ODE, який об'єднує початковий стан, спряження та інші часткові похідні в один вектор.

Більшість розв'язувачів ODE мають опцію виведення стану $z(t)$ у кількох точках часу. Коли втрати залежать від цих проміжних станів, зворотна похідна повинна бути розбита на послідовність окремих розв'язків, один між кожною парою вихідних часових точок. У кожному спостереженні спряження повинно бути скориговано у напрямку відповідної часткової похідної $\frac{\partial L}{\partial z(t_i)}$.

5.6.3. Метод очищення DiffPure

Метод дифузійного очищення (або скорочено DiffPure) [152] передбачає додавання шуму до зображень, на які здійснено атаки, відповідно до прямого процесу дифузійних моделей для отримання дифузованих зображень, з яких чисті зображення відновлюються через зворотний процес. Крім того, для цього методу надаються деякі теоретичні обґрунтування. Далі, застосовується сполучений метод для зворотного поширення через СДР для ефективної оцінки градієнтів при сильних адаптивних атаках.

Оскільки роль прямого СДР в рівн. (5.6.1) полягає в тому, щоб поступово видаляти локальні структури даних шляхом додавання шуму, ми припускаємо, що, маючи приклад атаки x_a, якщо ми розпочнемо прямий процес з $x(0) = x_a$, то атакуючі збурення, які є формою малих локальних структур, доданих до даних, також поступово згладжуються. Автори методу DiffPure приводять теорему і доведення, яка стверджує що KL-дивергенція між чистим та адверсаріальним розподілами даних монотонно зменшується при переході від t=0 до t=1 через прямий СДР [152].

Авторами DiffPure [152] було запропоновано двоетапний підхід, очищення від атак за допомогою дифузійних моделей: Маючи приклад атаки x_a на момент часу t=0, *тобто* $x(0) = x_a$, спочатку вирішується пряме СДР в рівн. (5.6.1) від t=0 до t=t^*, додаючи шум до вхідного зображення. Для VP-SDE, зашумлений зразок атаки на кроку дифузії $t^* \in [0, 1]$ можна ефективно вибрати за допомогою:

$$x(t^*) = \sqrt{\alpha(t^*)}x_a + \sqrt{1 - \alpha(t^*)} \tag{5.6.10}$$

де $\alpha(t) = e^{-\int_0^t \beta(s)ds}$ і $\sim \mathcal{N}(,_d)$.

По-друге, вирішується СДР зворотного часу в рівн. (5.6.2) від моменту часу t=t^*, використовуючи зашумлений зразок атаки $x(t^*)$, поданий в рівн. (5.6.10), як початкове значення для отримання кінцевого розв'язку $\hat{x}(0)$ СДР в рівн. (5.6.2). Оскільки $\hat{x}(0)$ не має аналітичного розв'язку, в методі DiffPure

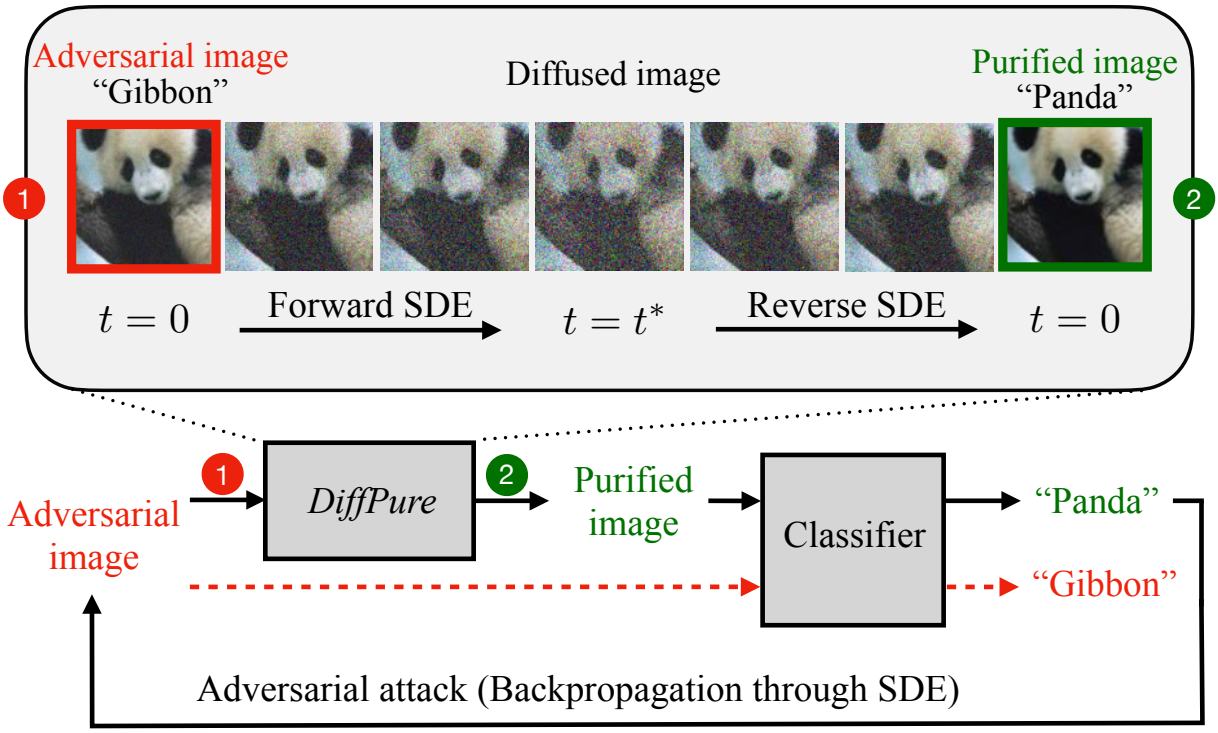


Рис. 5.9: Ілюстрація алгоритму *DiffPure* [152].

[152] використовується розв'язувач СДР `sdeint` (зазвичай з дискретизацією Ейлера-Маруями [162]). Тобто,

$$\hat{x}(0) = \texttt{sdeint}(x(t^*), f_{\text{rev}}, g_{\text{rev}}, \bar{w}, t^*, 0) \tag{5.6.11}$$

де `sdeint` визначено для послідовного прийому шести вхідних значень: початкове значення, коефіцієнт дрейфу, коефіцієнт дифузії, процес Вінера, початковий час і кінцевий час. Крім того, вищезазначені коефіцієнти дрейфу і дифузії визначаються як

$$\begin{aligned} f_{\text{rev}}(x,t) &:= -\frac{1}{2}\beta(t)[x + 2_\theta(x,t)] \\ g_{\text{rev}}(t) &:= \sqrt{\beta(t)} \end{aligned} \tag{5.6.12}$$

Отримані очищені дані $\hat{x}(0)$ потім передаються зовнішньому стандартному класифікатору для здійснення прогнозів. Ілюстрація даного методу показана на Рисунку 5.9.

Вибір кроку дифузії t^*: Крок повинен бути досить великим, щоб видалити локальні збурення від атак. Однак, t^* не може бути надмірно великим, оскільки глобальна семантика міток також буде видалена процесом дифузії,

якщо t^* продовжить збільшуватися. В результаті очищений зразок $\hat{x}(0)$ не зможе бути правильно класифікований. Формально, автори методу DiffPure [152] приводять теорему, яка характеризує, як крок дифузії t^* впливає на різницю між чистим зображенням x_0 і очищеним зображенням $\hat{x}(0)$.

Теорема 5.6.1. *Якщо ми припускаємо, що функція оцінки задовольняє умову* $\|s_\theta(x,t)\| \leqslant \frac{1}{2}C_s$*, тоді L2-відстань між чистими даними* x *та очищеними даними* $\hat{x}(0)$*, отриманими за рівн. (5.6.11), задовольняє умову з імовірністю щонайменше* $1-\delta$*, ми маємо*

$$\|\hat{x}(0) - x\| \leqslant \|_a\| + \sqrt{e^{2\gamma(t^*)} - 1}C_\delta + \gamma(t^*)C_s$$

де $_a$ *позначає атакуюче збурення, яке задовольняє* $x_a = x +_a$*,* $\gamma(t^*) := \int_0^{t^*} \frac{1}{2}\beta(s)ds$ *та константа* $C_\delta := \sqrt{2d + 4\sqrt{d\log\frac{1}{\delta}} + 4\log\frac{1}{\delta}}$.

Оскільки $\gamma(t^*)$ монотонно зростає з t^* і $\gamma(t^*) \geqslant 0$ для всіх t^*, останні два члени у вищезазначеній верхній межі також зростають з t^*. Таким чином, щоб зробити $\|\hat{x}(0) - x\|$ якомога меншим, t^* повинен бути достатньо малим. У крайніх випадках, коли t^*=0, ми маємо рівність $\|\hat{x}(0) - x\| = \|_a\|$, що означає, що $\hat{x}(0)$ зводиться до x_a, якщо ми не виконуємо дифузійне очищення.

Через компроміс між очищенням локальних збурень (при більшому t^*) і збереженням глобальних структур (при меншому t^*) атакуючих прикладів, існує оптимальний момент часу для кроку дифузії t^*, щоб отримати високу точність класифікації. Оскільки атакуючі збурення зазвичай малі, що можна усунути при малому t^*, найкращий t^* у більшості задач, пов'язаних з стійкістю до атак, також залишається відносно малим.

Адаптивні атаки на дифузійне очищення Сильні адаптивні атаки [13, 163] вимагають обчислення повних градієнтів системи захисту. Однак, просте зворотне поширення через розв'язувач СДР в рівн. (5.6.11) погано масштабується в обчислювальній пам'яті. Зокрема, позначимо N кількість обчислень функції при розв'язанні СДР, необхідна пам'ять зростає на $\mathcal{O}(N)$. Ця проблема ускладнює ефективну оцінку методу DiffPure [152] при силь-

них адаптивних атаках.

Попередні методи очищення від атак [164,165] страждають від тієї ж проблеми з пам'яттю при сильних адаптивних атаках. Таким чином, вони або оцінюють тільки за допомогою атак чорного ящика, або змінюють стратегію оцінки, щоб обійти обчислення повного градієнту (*наприклад*, використовуючи приблизні градієнти). Це ускладнює порівняння їх з методами навчання зі стійкістю до атак за більш стандартними протоколами оцінки (*наприклад*, AutoAttack). Щоб подолати це, було запропоновано використовувати *сполучений метод* [166] для ефективного обчислення повних градієнтів СДР без проблем з пам'яттю. Ідея полягає в тому, що градієнт через СДР можна отримати, розв'язавши інше розширене СДР [152].

Наступна твердження надає розширене СДР для обчислення градієнту об'єктиву по відношенню до вхідного $x(t^*)$ СДР в рівн. (5.6.11).

Твердження 5.6.2. *Для СДР в рівн. (5.6.11), розширене СДР, яке обчислює градієнт* $\frac{\partial}{\partial x(t^*)}$ *зворотного поширення через нього, задане як*

$$\begin{pmatrix} x(t^*) \\ \\ \frac{\partial}{\partial x(t^*)} \end{pmatrix} = \texttt{sdeint}\left(\begin{pmatrix} \hat{x}(0) \\ \\ \frac{\partial}{\partial \hat{x}(0)} \end{pmatrix}, \tilde{f}, \tilde{g}, \tilde{w}, 0, t^*\right) \tag{5.6.13}$$

де $\frac{\partial \mathcal{L}}{\partial \hat{x}(0)}$ *є градієнтом функції втрат* $\mathcal{L}$ *по відношенню до виходу* $\hat{x}(0)$ *СДР в рівн. (5.6.11), і*

$$\tilde{f}([x;z],t) = \begin{pmatrix} f_{rev}(x,t) \\ \\ \frac{\partial f_{rev}(x,t)}{\partial x} z \end{pmatrix}$$

$$\tilde{g}(t) = \begin{pmatrix} -g_{rev}(t) 1_d \\ 0_d \end{pmatrix}$$

$$\tilde{w}(t) = \begin{pmatrix} -w(1-t) \\ -w(1-t) \end{pmatrix}$$

де 1_d *та* 0_d *представляють* d*-вимірні вектори з усіма одиницями та всіма нулями відповідно.*

Ідеально, якщо розв'язувач СДР має малу числову помилку, градієнт, отриманий з цього твердження, буде тісно відповідати його справжньому значенню. Оскільки обчислення градієнту було перетворено на розв'язання розширеного СДР в рівн. (5.6.13), не потрібно зберігати проміжні операції і, таким чином, витрати пам'яті становлять $\mathcal{O}(1)$ [166]. Тобто, сполучений метод, описаний вище, перетворює СДР зворотного часу в рівн. (5.6.11) на диференційовану операцію (без проблем з пам'яттю). Оскільки крок прямої дифузії в рівн. (5.6.10) також є диференційованим за допомогою трюку репараметризації, можна легко обчислити повні градієнти функції втрат щодо атакуючих зображень для сильних адаптивних атак [152].

5.6.4. Аналіз стійкості систем з адверсаріальним тренуванням і очищенням

У цьому розділі представлені результати адверсаріальної точності для моделей, які були адверсаріально натреновані та захищені методом дифузійного очищення. Адверсаріальна точність оцінювалася за допомогою рандомізованого алгоритму AutoAttack (APGD-CE та APGD-DLR) під загрозою L_∞ без перезапуску і з 5 ітераціями EoT (Expectation over Transformation). Бюджет на збурення ϵ було встановлено на рівні 4/255.

Були використані ті самі трансформерні моделі, що і в попередньому експерименті: базову ViT та ViT з MEGA блоком та активаційною функцією RAF2d. У таблиці нижче наведені результати адверсаріальної точності моделей з та без дифузійного очищення.

Для розрахунку адверсаріальної точності з очищенням було використано тільки 160 зображень з тестової вибірки датасету, оскільки обчислення градієнту зворотнього дифузійного процесу займало досить багато обчислень.

Таблиця 5.3

Адверсаріальна і чиста точність під загрозою L_∞ з використанням рандомізованого AutoAttack (APGD-CE та APGD-DLR)

Модель	**Захист**	**адв. точність**	**точність**
ViT	Без очищення	**12.5%**	**24.1%**
ViT	З очищенням	**15.17%**	**23.6%**
ViT + MEGA + RAF2d	Без очищення	0.8%	19.5%
ViT + MEGA + RAF2d	З очищенням	7.29%	15.6%

Як показано в таблиці, адверсаріальна точність оцінювалася як без дифузійного очищення, так і з очищенням. Це дозволяє оцінити ефективність методу очищення для покращення стійкості моделей до адверсаріальних атак.

Ці результати показують, що нейронна мережа ViT разом з блоком MEGA та розглянутою багатовимірною функцією активацію значно поступається в точності звичайній моделі ViT, незважаючи на те що дана архітектура показувала кращі результати класифікації для тренування без адверсаріальних прикладів. Даний результат свідчить про те, що для розглянутої параметризації ViT необхідний додатковий підбір оптимальних гіперпараметрів і налаштувань для адверсаріального тренування, щоб розкрити повний потенціал, який спостерігався в 5.3 розділі. Також, можна переконатись що дифузійне очищення значно покращує адверсаріальну точність, хоч і за рахунок незначного погіршення класифікації звичайних зображень. Потенційно, цю різницю між звичайною і адверсаріальною точністю можна нівелювати за рахунок додаткового тренування моделі класифікатора на ’знешумлених’ зображеннях дифузійної моделі. Недоліком такого підходу є надзвичайно висока кількість обчислювального часу, яка необхідна щоб згенерувати знешумлені зображення для такого тренування, тож це залишається для майбутніх досліджень.

ВИСНОВКИ

Дослідження, проведені в рамках цієї дисертації, охоплювали три важливі експерименти, спрямовані на вдосконалення нейро-мережевих моделей з архітектурою ViT та їх оцінку з точки зору стандартної та адверсаріальної точності. Підсумкові результати експериментів наведені нижче.

У першому експерименті були натреновані моделі ViT, оснащені з допомогою модулю MEGA та мультіваріативних активаційних функцій, і порівняні з базовою моделлю. Отримані результати показали, що вдосконалені моделі продемонстрували кращу точність класифікації, а адверсаріальна точність була на рівні або трохи кращою, ніж у базової моделі.

У другому експерименті моделі, оснащені з допомогою MEGA та мультіваріативних активаційних функцій, були натреновані з використанням адверсаріального тренування та порівняні з базовою моделлю, яка теж тренувалась адверсаріально. Також, для тренування були використані метод аугментації mixup та згладжена loss-функція крос ентропії. Результати показали, що розглянуті моделі мали гіршу як стандартну, так і адверсаріальну точність. Це пояснюється нестабільністю тренування. Ця нестабільність була частково зменшена за допомогою експоненціального середнього згладжування оновлених параметрів моделі, але для стабілізації адверсаріального тренування розглянутих параметризацій необхідні додаткові методи.

У третьому експерименті були порівняні моделі з другого експерименту з базовою, і, як очікувалося, базові моделі показали кращу точність як з дифузійним очищенням, так і без нього. Це знову-таки пояснюється нестабільністю тренування. Окрім цього, дифузійні моделі значно покращили адверсаріальну стійкість, незважаючи на незначне зменшення точності класифікації. Було висунуте припущення, що розрив у точності класифікації може бути

усунений шляхом тренування класифікатора на дифузійно-очищених зображеннях. Це залишено для майбутніх експериментів.

Таким чином, проведені дослідження підтверджують ефективність використання розглянутих нейро-мережевих моделей для підвищення точності класифікації та демонструють складності які виникають при адверсаріальному тренуванні.

Список використаних джерел

1. Zhang Hongyang, Yu Yaodong, Jiao Jiantao, Xing Eric, Ghaoui Laurent El, and Jordan Michael. Theoretically Principled Trade-off between Robustness and Accuracy // Proceedings of the 36th International Conference on Machine Learning / ed. by Chaudhuri Kamalika and Salakhutdinov Ruslan. — PMLR. — 2019. — 09–15 Jun. — Vol. 97 of Proceedings of Machine Learning Research. — P. 7472–7482. — Access mode: https://proceedings.mlr.press/v97/zhang19p.html.
2. Ma Xuezhe, Zhou Chunting, Kong Xiang, He Junxian, Gui Liangke, Neubig Graham, May Jonathan, and Zettlemoyer Luke. Mega: Moving Average Equipped Gated Attention. — 2022. — Access mode: https://arxiv.org/abs/2209.10655.
3. Szegedy Christian, Zaremba Wojciech, Sutskever Ilya, Bruna Joan, Erhan Dumitru, Goodfellow Ian and Fergus Rob. Intriguing properties of neural networks // International Conference on Learning Representations. — 2014. — Jan. — 2nd International Conference on Learning Representations, ICLR 2014. online; accessed: http://arxiv.org/abs/1312.6199.
4. Goodfellow Ian, Shlens Jonathon, and Szegedy Christian. Explaining and Harnessing Adversarial Examples // arXiv:1412.6572. — 2014. — Dec.
5. Papernot Nicolas, Mcdaniel Patrick, Jha Somesh, Fredrikson Matt, Celik Z. Berkay and Swami Ananthram. The limitations of deep learning in adversarial settings // Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016. — United States : Institute of Electrical and Electronics Engineers Inc. — 2016. — May. — Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016. — P. 372–387. — 1st IEEE European Symposium on Security and Privacy,

EURO S and P 2016 ; Conference date: 21-03-2016 Through 24-03-2016.

6. Kurakin Alexey, Goodfellow Ian, and Bengio Samy. Adversarial examples in the physical world // arXiv:1607.02533. — 2016. — July.
7. Ivanyuk-Skulskiy B., Kriukova G., and Dmytryshyn A. Geometric Properties of Adversarial Images // 2020 IEEE Third International Conference on Data Stream Mining Processing (DSMP). — 2020.
8. Li Yao, Cheng Minhao, Hsieh Cho-Jui, and Lee Thomas Chun Man. A Review of Adversarial Attack and Defense for Classification Methods // The American Statistician. — 2021. — Vol. 76. — P. 329 – 345. — Access mode: https://api.semanticscholar.org/CorpusID:244419907.
9. Croce Francesco and Hein Matthias. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks // Proceedings of the 37th International Conference on Machine Learning / ed. by III Hal Daumé and Singh Aarti. — PMLR. — 2020. — 13–18 Jul. — Vol. 119 of Proceedings of Machine Learning Research. — P. 2206–2216. — Access mode: https://proceedings.mlr.press/v119/croce20b.html.
10. Yang Puyudi, Chen Jianbo, Hsieh Cho-Jui, Wang Jane-Ling, and Jordan Michael. Ml-loo: Detecting adversarial examples with feature attribution // Proceedings of the AAAI Conference on Artificial Intelligence. — 2020. — Vol. 34. — P. 6639–6647.
11. Madry Aleksander, Makelov Aleksandar, Schmidt Ludwig, Tsipras Dimitris, and Vladu Adrian. Towards Deep Learning Models Resistant to Adversarial Attacks // International Conference on Learning Representations. — 2018.
12. Carlini Nicholas and Wagner David A. Towards Evaluating the Robustness of Neural Networks // 2017 IEEE Symposium on Security and Privacy (SP). — 2017. — P. 39–57.
13. Athalye Anish, Carlini Nicholas, and Wagner David. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples // International conference on machine learning / PMLR. — 2018. —

P. 274–283.

14. Mosbach Marius, Andriushchenko Maksym, Trost Thomas Alexander, Hein Matthias, and Klakow Dietrich. Logit Pairing Methods Can Fool Gradient-Based Attacks // ArXiv. — 2018. — Vol. abs/1810.12042. — Access mode: https://api.semanticscholar.org/CorpusID:53104902.
15. Carlini Nicholas, Athalye Anish, Papernot Nicolas, Brendel Wieland, Rauber Jonas, Tsipras Dimitris, Goodfellow Ian, Madry Aleksander, and Kurakin Alexey. On evaluating adversarial robustness // arXiv preprint arXiv:1902.06705. — 2019.
16. Carlini Nicholas and Wagner David. Adversarial examples are not easily detected: Bypassing ten detection methods // Proceedings of the 10th ACM workshop on artificial intelligence and security. — 2017. — P. 3–14.
17. Zou Hui and Hastie Trevor. Regularization and variable selection via the elastic net // Journal of the Royal Statistical Society: Series B. — 2005. — Vol. 67, no. 2. — P. 301–320.
18. Chen Pin-Yu, Sharma Yash, Zhang Huan, Yi Jinfeng, and Hsieh Cho-Jui. EAD: elastic-net attacks to deep neural networks via adversarial examples // Thirty-second AAAI conference on artificial intelligence. — 2018.
19. Moosavi-Dezfooli Seyed-Mohsen, Fawzi Alhussein, and Frossard Pascal. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — 06. — P. 2574–2582.
20. Wong Eric, Schmidt Frank, and Kolter Zico. Wasserstein adversarial examples via projected sinkhorn iterations // International Conference on Machine Learning. — 2019. — P. 6808–6817.
21. Wong Eric and Kolter J Zico. Learning perturbation sets for robust machine learning // arXiv preprint arXiv:2007.08450. — 2020.
22. Croce Francesco and Hein Matthias. Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack // Proceedings of the 37th

International Conference on Machine Learning / ed. by III Hal Daumé and Singh Aarti. — PMLR. — 2020. — 13–18 Jul. — Vol. 119 of Proceedings of Machine Learning Research. — P. 2196–2205. — Access mode: https://proceedings.mlr.press/v119/croce20a.html.

23. Chen Pin-Yu, Zhang Huan, Sharma Yash, Yi Jinfeng, and Hsieh Cho-Jui. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models // Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. — 2017. — P. 15–26. — hep-ph/9609357.
24. Ilyas Andrew, Engstrom Logan, Athalye Anish, and Lin Jessy. Black-box adversarial attacks with limited queries and information // International Conference on Machine Learning. — 2018. — P. 2137–2146.
25. Deng Jia, Dong Wei, Socher Richard, Li Li-Jia, Li Kai, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database // Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. — 2009. — P. 248–255. — hep-ph/9609357.
26. Ilyas Andrew, Engstrom Logan, and Madry Aleksander. Prior Convictions: Black-box Adversarial Attacks with Bandits and Priors // International Conference on Learning Representations. — 2019.
27. Liu Sijia, Chen Pin-Yu, Chen Xiangyi, and Hong Mingyi. signSGD via zeroth-order oracle // International Conference on Learning Representations. — 2018.
28. Al-Dujaili Abdullah and O'Reilly Una-May. Sign Bits Are All You Need for Black-Box Attacks // International Conference on Learning Representations. — 2020.
29. Guo Yiwen, Yan Ziang, and Zhang Changshui. Subspace Attack: Exploiting Promising Subspaces for Query-Efficient Black-box Attacks // Advances in Neural Information Processing Systems / ed. by Wallach H., Larochelle H., Beygelzimer A., d Alché-Buc F., Fox E., and Garnett R. — Curran Asso-

ciates, Inc. — 2019. — Vol. 32.

30. Cheng Minhao, Le Thong, Chen Pin-Yu, Zhang Huan, Yi JinFeng, and Hsieh Cho-Jui. Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach // International Conference on Learning Representations. — 2019. — Access mode: https://openreview.net/forum?id=rJlk6iRqKX.
31. Wang Lu, Zhang Huan, Yi Jinfeng, Hsieh Cho-Jui, and Jiang Yuan. Spanning attack: reinforce black-box attacks with unlabeled data // Machine Learning. — 2020. — Vol. 109, no. 12. — P. 2349–2368.
32. Andriushchenko Maksym, Croce Francesco, Flammarion Nicolas, and Hein Matthias. Square attack: a query-efficient black-box adversarial attack via random search // European Conference on Computer Vision. — 2020. — P. 484–501.
33. Li Yandong, Li Lijun, Wang Liqiang, Zhang Tong, and Gong Boqing. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks // International Conference on Machine Learning / PMLR. — 2019. — P. 3866–3876.
34. Alzantot Moustafa, Sharma Yash, Chakraborty Supriyo, and Srivastava Mani. Genattack: Practical black-box attacks with gradient-free optimization // arXiv preprint arXiv:1805.11090. — 2018.
35. Guo Chuan, Gardner Jacob, You Yurong, Wilson Andrew Gordon, and Weinberger Kilian. Simple Black-box Adversarial Attacks // Proceedings of the 36th International Conference on Machine Learning / ed. by Chaudhuri Kamalika and Salakhutdinov Ruslan. — PMLR. — 2019. — 09–15 Jun. — Vol. 97 of Proceedings of Machine Learning Research. — P. 2484–2493.
36. Papernot Nicolas, McDaniel Patrick, Goodfellow Ian, Jha Somesh, Celik Z Berkay, and Swami Ananthram. Practical black-box attacks against machine learning // Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. — 2017. — P. 506–519. — hep-

ph/9609357.

37. Brendel Wieland, Rauber Jonas, and Bethge Matthias. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models // International Conference on Learning Representations. — 2018.
38. Brunner Thomas, Diehl Frederik, Le Michael Truong, and Knoll Alois. Guessing smart: Biased sampling for efficient black-box adversarial attacks // Proceedings of the IEEE International Conference on Computer Vision. — 2019. — P. 4958–4966.
39. Chen Jianbo and Jordan Michael I. Boundary attack++: Query-efficient decision-based adversarial attack // arXiv preprint arXiv:1904.02144. — 2019.
40. Chen Jianbo, Jordan Michael I, and Wainwright Martin J. Hopskipjumpattack: A query-efficient decision-based attack // 2020 IEEE Symposium on Security and Privacy (SP). — 2020. — P. 1277–1294.
41. Guo Chuan, Frank Jared S., and Weinberger Kilian Q. Low Frequency Adversarial Perturbation // Proceedings of the 35th Uncertainty in Artificial Intelligence Conference / ed. by Adams Ryan P. and Gogate Vibhav. — Tel Aviv, Israel : PMLR. — 2020. — 22–25 Jul. — Vol. 115 of Proceedings of Machine Learning Research. — P. 1127–1137.
42. Cheng Minhao, Singh Simranjit, Chen Patrick H., Chen Pin-Yu, Liu Sijia, and Hsieh Cho-Jui. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack // International Conference on Learning Representations. — 2020.
43. Liu Yanpei, Chen Xinyun, Liu Chang, and Song Dawn. Delving into transferable adversarial examples and black-box attacks // International Conference on Learning Representations. — 2017.
44. Cheng Shuyu, Dong Yinpeng, Pang Tianyu, Su Hang, and Zhu Jun. Improving black-box adversarial attacks with a transfer-based prior // Advances in Neural Information Processing Systems. — 2019. — P. 10932–10942.

45. Nesterov Yurii and Spokoiny Vladimir. Random gradient-free minimization of convex functions // Foundations of Computational Mathematics. — 2017. — Vol. 17, no. 2. — P. 527–566.
46. Chen Jinghui and Gu Quanquan. Rays: A ray searching method for hard-label adversarial attack // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. — 2020. — P. 1739–1747.
47. Su Jiawei, Vargas Danilo Vasconcellos, and Sakurai Kouichi. One Pixel Attack for Fooling Deep Neural Networks // IEEE Transactions on Evolutionary Computation. — 2019. — Oct. — Vol. 23, no. 5. — P. 828–841. — Access mode: http://dx.doi.org/10.1109/TEVC.2019.2890858.
48. Brown Tom, Mane Dandelion, Roy Aurko, Abadi Martin, and Gilmer Justin. Adversarial Patch // NIPS Workshop (2017). — 2017. — 1712.09665.
49. Aldahdooh Ahmed, Hamidouche Wassim, Fezza Sid Ahmed, and Déforges Olivier. Adversarial example detection for DNN models: a review and experimental comparison // Artificial Intelligence Review. — 2021. — Vol. 55. — P. 4403 – 4462. — Access mode: https://api.semanticscholar.org/CorpusID:233481928.
50. Ilyas Andrew, Santurkar Shibani, Tsipras Dimitris, Engstrom Logan, Tran Brandon, and Madry Aleksander. Adversarial Examples Are Not Bugs, They Are Features // Advances in Neural Information Processing Systems / ed. by Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., and Garnett R. — Curran Associates, Inc. — 2019. — Vol. 32. — Access mode: https://proceedings.neurips.cc/paper_files/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.
51. Srivastava Nitish, Hinton Geoffrey, Krizhevsky Alex, Sutskever Ilya, and Salakhutdinov Ruslan. Dropout: a simple way to prevent neural networks from overfitting // The journal of machine learning research. — 2014. — Vol. 15, no. 1. — P. 1929–1958.

52. Feinman Reuben, Curtin Ryan R., Shintre Saurabh, and Gardner Andrew B. Detecting Adversarial Samples from Artifacts // CoRR. — 2017. — Vol. abs/1703.00410. — 1703.00410.
53. Smith Lewis and Gal Yarin. Understanding Measures of Uncertainty for Adversarial Example Detection // Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018 / ed. by Globerson Amir and Silva Ricardo. — AUAI Press. — 2018. — P. 560–569. — Access mode: http://auai.org/uai2018/proceedings/papers/207.pdf.
54. Hendrycks Dan and Gimpel Kevin. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks // 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. — OpenReview.net. — 2017. — Access mode: https://openreview.net/forum?id=Hkg4TI9xl.
55. Pertigkiozoglou Stefanos and Maragos Petros. Detecting Adversarial Examples in Convolutional Neural Networks // CoRR. — 2018. — Vol. abs/1812.03303. — 1812.03303.
56. Aigrain Jonathan and Detyniecki Marcin. Detecting Adversarial Examples and Other Misclassifications in Neural Networks by Introspection // CoRR. — 2019. — Vol. abs/1905.09186. — 1905.09186.
57. Monteiro João, Albuquerque Isabela, Akhtar Zahid, and Falk Tiago H. Generalizable adversarial examples detection based on bi-model decision mismatch // 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) / IEEE. — 2019. — P. 2839–2844.
58. Gong Zhitao, Wang Wenlu, and Ku Wei-Shinn. Adversarial and Clean Data Are Not Twins // CoRR. — 2017. — Vol. abs/1704.04960. — 1704.04960.
59. Grosse Kathrin, Manoharan Praveen, Papernot Nicolas, Backes Michael, and McDaniel Patrick D. On the (Statistical) Detection of Adversarial Examples // CoRR. — 2017. — Vol. abs/1702.06280. — 1702.06280.

60. Hosseini Hossein, Chen Yize, Kannan Sreeram, Zhang Baosen, and Poovendran Radha. Blocking Transferability of Adversarial Examples in Black-Box Learning Systems // CoRR. — 2017. — Vol. abs/1703.04318. — 1703.04318.
61. Szegedy Christian, Vanhoucke Vincent, Ioffe Sergey, Shlens Jonathon, and Wojna Zbigniew. Rethinking the inception architecture for computer vision. 2015 // arXiv preprint arXiv:1512.00567. — 2015.
62. Kherchouche Anouar, Fezza Sid Ahmed, Hamidouche Wassim, and Déforges Olivier. Detection of Adversarial Examples in Deep Neural Networks with Natural Scene Statistics // 2020 International Joint Conference on Neural Networks (IJCNN) / IEEE. — 2020. — P. 1–7.
63. Mittal Anish, Moorthy Anush Krishna, and Bovik Alan Conrad. No-reference image quality assessment in the spatial domain // IEEE Transactions on image processing. — 2012. — Vol. 21, no. 12. — P. 4695–4708.
64. Lust Julia and Condurache Alexandru Paul. GraN: An Efficient Gradient-Norm Based Detector for Adversarial and Misclassified Examples // 28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2020, Bruges, Belgium, October 2-4, 2020. — 2020. — P. 7–12. — Access mode: https://www.esann.org/sites/default/files/proceedings/2020/ES2020-159.pdf.
65. Zuo Fei and Zeng Qiang. Exploiting the Sensitivity of L2 Adversarial Examples to Erase-and-Restore // ASIA CCS ’21: ACM Asia Conference on Computer and Communications Security, Virtual Event, Hong Kong, June 7-11, 2021 / ed. by Cao Jiannong, Au Man Ho, Lin Zhiqiang, and Yung Moti. — ACM. — 2021. — P. 40–51. — Access mode: https://doi.org/10.1145/3433210.3437529.
66. Gretton Arthur, Borgwardt Karsten M, Rasch Malte J, Schölkopf Bernhard, and Smola Alexander. A kernel two-sample test // The Journal of Machine Learning Research. — 2012. — Vol. 13, no. 1. — P. 723–773.

67. Li Xin and Li Fuxin. Adversarial examples detection in deep networks with convolutional filter statistics // Proceedings of the IEEE International Conference on Computer Vision. — 2017. — P. 5764–5772.
68. Ma Xingjun, Li Bo, Wang Yisen, Erfani Sarah M., Wijewickrema Sudanthi N. R., Schoenebeck Grant, Song Dawn, Houle Michael E., and Bailey James. Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality // 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. — OpenReview.net. — 2018. — Access mode: https://openreview.net/forum?id=B1gJ1L2aW.
69. Karger David and Ruhl Matthias. Finding Nearest Neighbors in Growth-restricted Metrics // Conference Proceedings of the Annual ACM Symposium on Theory of Computing. — 2002. — 09.
70. Houle Michael E., Kashima Hisashi, and Nett Michael. Generalized Expansion Dimension // 2012 IEEE 12th International Conference on Data Mining Workshops. — 2012. — P. 587–594.
71. Houle Michael E. Local Intrinsic Dimensionality I: An Extreme-Value-Theoretic Foundation for Similarity Applications // Similarity Search and Applications / ed. by Beecks Christian, Borutta Felix, Kröger Peer, and Seidl Thomas. — Cham : Springer International Publishing. — 2017. — P. 64–79.
72. Amsaleg Laurent, Chelly Oussama, Furon Teddy, Girard Stéphane, Houle Michael E., ichi Kawarabayashi Ken, and Nett Michael. Estimating Local Intrinsic Dimensionality // Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2015. — Access mode: https://api.semanticscholar.org/CorpusID:16058196.
73. Lee Kimin, Lee Kibok, Lee Honglak, and Shin Jinwoo. A simple unified framework for detecting out-of-distribution samples and adversarial attacks //

Advances in Neural Information Processing Systems. — 2018. — P. 7167–7177.

74. Cohen Gilad, Sapiro Guillermo, and Giryes Raja. Detecting adversarial samples using influence functions and nearest neighbors // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2020. — P. 14453–14462.
75. Mao Xiaofeng, Chen Yuefeng, Li Yuhong, He Yuan, and Xue Hui. Learning to Characterize Adversarial Subspaces // ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) / IEEE. — 2020. — P. 2438–2442.
76. Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N., Kaiser Lukasz, and Polosukhin Illia. Attention is All you Need // Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. — 2017. — P. 5998–6008. — Access mode: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.
77. Lu Jiajun, Issaranon Theerasit, and Forsyth David. Safetynet: Detecting and rejecting adversarial examples robustly // Proceedings of the IEEE International Conference on Computer Vision. — 2017. — P. 446–454.
78. Metzen Jan Hendrik, Genewein Tim, Fischer Volker, and Bischoff Bastian. On Detecting Adversarial Perturbations // 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. — OpenReview.net. — 2017. — Access mode: https://openreview.net/forum?id=SJzCSf9xg.
79. Carrara Fabio, Becarelli Rudy, Caldelli Roberto, Falchi Fabrizio, and Amato Giuseppe. Adversarial examples detection in features distance spaces // Proceedings of the European Conference on Computer Vision (ECCV). — 2018. — P. 0–0.

80. Eniser Hasan Ferit, Christakis Maria, and Wüstholz Valentin. RAID: Randomized Adversarial-Input Detection for Neural Networks // CoRR. — 2020. — Vol. abs/2002.02776. — 2002.02776.
81. Carrara Fabio, Falchi Fabrizio, Caldelli Roberto, Amato Giuseppe, Fumarola Roberta, and Becarelli Rudy. Detecting adversarial example attacks to deep neural networks // Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing. — 2017. — P. 1–7.
82. Pang Tianyu, Du Chao, Dong Yinpeng, and Zhu Jun. Towards robust detection of adversarial examples // Advances in Neural Information Processing Systems. — 2018. — P. 4579–4589.
83. Sotgiu Angelo, Demontis Ambra, Melis Marco, Biggio Battista, Fumera Giorgio, Feng Xiaoyi, and Roli Fabio. Deep neural rejection against adversarial examples // EURASIP Journal on Information Security. — 2020. — Vol. 2020. — P. 1–10.
84. Aldahdooh Ahmed, Hamidouche Wassim, and Déforges Olivier. Revisiting Model's Uncertainty and Confidences for Adversarial Example Detection // arXiv preprint arXiv:2103.05354. — 2021.
85. Geifman Yonatan and El-Yaniv Ran. SelectiveNet: A Deep Neural Network with an Integrated Reject Option // CoRR. — 2019. — Vol. abs/1901.09192. — 1901.09192.
86. Hendrycks Dan and Gimpel Kevin. Early Methods for Detecting Adversarial Images // 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. — OpenReview.net. — 2017. — Access mode: https://openreview.net/forum?id=B1dexpDug.
87. Zheng Zhihao and Hong Pengyu. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks // Advances in Neural Information Processing Systems. — 2018. — P. 7913–7922.
88. Miller David, Wang Yujia, and Kesidis George. When Not to Classify:

Anomaly Detection of Attacks (ADA) on DNN classifiers at test time // Neural computation. — 2019. — Vol. 31, no. 8. — P. 1624–1670.

89. Song Yang, Kim Taesup, Nowozin Sebastian, Ermon Stefano, and Kushman Nate. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples // 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. — OpenReview.net. — 2018. — Access mode: https://openreview.net/forum?id=rJUYGxbCW.
90. Van den Oord Aaron, Kalchbrenner Nal, Espeholt Lasse, Vinyals Oriol, Graves Alex, et al. Conditional image generation with pixelcnn decoders // Advances in neural information processing systems. — 2016. — Vol. 29. — P. 4790–4798.
91. Meng Dongyu and Chen Hao. Magnet: a two-pronged defense against adversarial examples // Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. — 2017. — P. 135–147.
92. Xu Weilin, Evans David, and Qi Yanjun. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks // 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018. — The Internet Society. — 2018. — Access mode: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03A-4_Xu_paper.pdf.
93. Liang Bin, Li Hongcheng, Su Miaoqiang, Li Xirong, Shi Wenchang, and Wang Xiaofeng. Detecting Adversarial Image Examples in Deep Neural Networks with Adaptive Noise Reduction // IEEE Trans. Dependable Secur. Comput. — 2021. — Vol. 18, no. 1. — P. 72–85. — Access mode: https://doi.org/10.1109/TDSC.2018.2874243.
94. Ma Shiqing and Liu Yingqi. Nic: Detecting adversarial samples with neural network invariant checking // Proceedings of the 26th Network and Distributed System Security Symposium (NDSS 2019). — 2019.

95. Freitas Scott, Chen Shang-Tse, Wang Zijie J., and Chau Duen Horng. UnMask: Adversarial Detection and Defense Through Robust Feature Alignment // IEEE International Conference on Big Data, Big Data 2020, Atlanta, GA, USA, December 10-13, 2020. — IEEE. — 2020. — P. 1081–1088. — Access mode: https://doi.org/10.1109/BigData50022.2020.9378303.
96. Blundell Charles, Cornebise Julien, Kavukcuoglu Koray, and Wierstra Daan. Weight Uncertainty in Neural Network // International Conference on Machine Learning. — 2015. — P. 1613–1622.
97. Li Yao, Tang Tongyi, Hsieh Cho-Jui, and Lee Thomas Chun Man. Adversarial Examples Detection With Bayesian Neural Network // IEEE Transactions on Emerging Topics in Computational Intelligence. — 2021. — Access mode: https://api.semanticscholar.org/CorpusID:234763286.
98. Carbone Ginevra, Wicker Matthew, Laurenti Luca, Patane' Andrea, Bortolussi Luca, and Sanguinetti Guido. Robustness of Bayesian Neural Networks to Gradient-Based Attacks // Advances in Neural Information Processing Systems / ed. by Larochelle H., Ranzato M., Hadsell R., Balcan M. F., and Lin H. — Curran Associates, Inc. — 2020. — Vol. 33. — P. 15602–15613.
99. Liu Xuanqing, Li Yao, Wu Chongruo, and Hsieh Cho-Jui. Adv-BNN: Improved Adversarial Defense through Robust Bayesian Neural Network // International Conference on Learning Representations. — 2019.
100. Ye Nanyang and Zhu Zhanxing. Bayesian Adversarial Learning // Advances in Neural Information Processing Systems 31 / ed. by Bengio S., Wallach H., Larochelle H., Grauman K., Cesa-Bianchi N., and Garnett R. — Curran Associates, Inc., 2018. — P. 6892–6901. — Access mode: http://papers.nips.cc/paper/7921-bayesian-adversarial-learning.pdf.
101. Zhang Jiani, Shi Xingjian, Xie Junyuan, Ma Hao, King Irwin, and Yeung Dit-Yan. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. — 2018. — Access mode: https://arxiv.org/abs/1803.07294.

102. McKenzie Eddie and Gardner Jr Everette S. Damped trend exponential smoothing: a modelling viewpoint // International Journal of Forecasting. — 2010. — Vol. 26, no. 4. — P. 661–665.
103. Svetunkov Ivan. Complex exponential smoothing. — Lancaster University (United Kingdom), 2016.
104. Oord Aaron van den, Vinyals Oriol, and Kavukcuoglu Koray. Neural Discrete Representation Learning. — 2017. — Access mode: https://arxiv.org/abs/1711.00937.
105. Radford Alec and Narasimhan Karthik. Improving Language Understanding by Generative Pre-Training. — 2018.
106. Elizalde Benjamin, Deshmukh Soham, Ismail Mahmoud Al, and Wang Huaming. CLAP Learning Audio Concepts from Natural Language Supervision // ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). — 2023. — P. 1–5.
107. Kingma Diederik P and Welling Max. Auto-Encoding Variational Bayes. — 2022. — 1312.6114.
108. Kong Jungil, Kim Jaehyeon, and Bae Jaekyoung. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. — 2020. — 2010.05646.
109. Liu Haohe, Tian Qiao, Yuan Yi, Liu Xubo, Mei Xinhao, Kong Qiuqiang, Wang Yuping, Wang Wenwu, Wang Yuxuan, and Plumbley Mark D. AudioLDM 2: Learning Holistic Audio Generation with Self-supervised Pretraining. — 2023. — 2308.05734.
110. Chen Sanyuan, Wang Chengyi, Chen Zhengyang, Wu Yu, Liu Shujie, Chen Zhuo, Li Jinyu, Kanda Naoyuki, Yoshioka Takuya, Xiao Xiong, Wu Jian, Zhou Long, Ren Shuo, Qian Yanmin, Qian Yao, Wu Jian, Zeng Michael, Yu Xiangzhan, and Wei Furu. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing // IEEE Journal of Selected Topics in Signal Processing. — 2022. — Đ□ĐºÑ□. — Vol. 16,

no. 6. — P. 1505–1518. — Access mode: http://dx.doi.org/10.1109/JSTSP.2022.3188113.

111. Ma Ziyang, Zheng Zhisheng, Ye Jiaxin, Li Jinchao, Gao Zhifu, Zhang Shiliang, and Chen Xie. emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation. — 2023. — 2312.15185.
112. Radford Alec, Kim Jong Wook, Xu Tao, Brockman Greg, McLeavey Christine, and Sutskever Ilya. Robust Speech Recognition via Large-Scale Weak Supervision. — 2022. — 2212.04356.
113. Maas Andrew L, Hannun Awni Y, and Ng Andrew Y. Rectifier nonlinearities improve neural network acoustic models // International Conference on Machine Learning. — 2013. — Vol. 30. — P. 3.
114. He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification // IEEE international conference on computer vision. — 2015. — P. 1026–1034.
115. Clevert Djork-Arné, Unterthiner Thomas, and Hochreiter Sepp. Fast and accurate deep network learning by exponential linear units (elus) // International Conference on Learning Representations. — 2016.
116. Hendrycks Dan and Gimpel Kevin. Gaussian error linear units (gelus) // arXiv preprint arXiv:1606.08415. — 2016.
117. Elfwing Stefan, Uchibe Eiji, and Doya Kenji. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning // Neural Networks. — 2018. — Vol. 107. — P. 3–11.
118. Misra Diganta. Mish: A Self Regularized Non-Monotonic Neural Activation Function // arXiv preprint arXiv:1908.08681. — 2019.
119. Meronen Lassi, Trapp Martin, and Solin Arno. Periodic Activation Functions Induce Stationarity // CoRR. — 2021. — Vol. abs/2110.13572. — arXiv : 2110.13572.

120. Shazeer Noam. GLU Variants Improve Transformer // CoRR. — 2020. — Vol. abs/2002.05202. — arXiv : 2002.05202.

121. Ma Ningning, Zhang Xiangyu, Liu Ming, and Sun Jian. Activate or Not: Learning Customized Activation // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2021. — June. — P. 8032–8042.

122. Molina Alejandro, Schramowski Patrick, and Kersting Kristian. Padé Activation Units: End-to-end Learning of Flexible Activation Functions in Deep Networks // International Conference on Learning Representations. — 2020.

123. Dosovitskiy Alexey, Beyer Lucas, Kolesnikov Alexander, Weissenborn Dirk, Zhai Xiaohua, Unterthiner Thomas, Dehghani Mostafa, Minderer Matthias, Heigold Georg, Gelly Sylvain, Uszkoreit Jakob, and Houlsby Neil. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale // ArXiv. — 2020. — Vol. abs/2010.11929. — Access mode: https://api.semanticscholar.org/CorpusID:225039882.

124. Cohen Jeremy, Rosenfeld Elan, and Kolter Zico. Certified Adversarial Robustness via Randomized Smoothing // Proceedings of the 36th International Conference on Machine Learning / ed. by Chaudhuri Kamalika and Salakhutdinov Ruslan. — PMLR. — 2019. — 09–15 Jun. — Vol. 97 of Proceedings of Machine Learning Research. — P. 1310–1320. — Access mode: https://proceedings.mlr.press/v97/cohen19c.html.

125. Croce Francesco, Andriushchenko Maksym, Sehwag Vikash, Debenedetti Edoardo, Flammarion Nicolas, Chiang Mung, Mittal Prateek, and Hein Matthias. Robustbench: a standardized adversarial robustness benchmark // arXiv preprint arXiv:2010.09670. — 2020.

126. Athalye Anish and Carlini Nicholas. On the Robustness of the CVPR 2018 White-Box Adversarial Example Defenses // CoRR. — 2018. — Vol. abs/1804.03286. — arXiv : 1804.03286.

127. Tramèr Florian, Kurakin Alexey, Papernot Nicolas, Boneh Dan, and Mc-

daniel Patrick. Ensemble Adversarial Training: Attacks and Defenses // ArXiv. — 2017. — Vol. abs/1705.07204. — Access mode: https://api.semanticscholar.org/CorpusID:21946795.

128. Huang Ruitong, Xu Bing, Schuurmans Dale, and Szepesvari Csaba. Learning with a Strong Adversary // ArXiv. — 2015. — Vol. abs/1511.03034. — Access mode: https://api.semanticscholar.org/CorpusID:16233336.
129. Shaham Uri, Yamada Yutaro, and Negahban Sahand N. Understanding adversarial training: Increasing local stability of supervised models through robust optimization // Neurocomputing. — 2015. — Vol. 307. — P. 195–204. — Access mode: https://api.semanticscholar.org/CorpusID:268118338.
130. Bai Tao, Luo Jinqi, Zhao Jun, Wen Bihan, and Wang Qian. Recent Advances in Adversarial Training for Adversarial Robustness // Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21 / ed. by Zhou Zhi-Hua. — International Joint Conferences on Artificial Intelligence Organization. — 2021. — 8. — P. 4312–4321. — Survey Track. Access mode: https://doi.org/10.24963/ijcai.2021/591.
131. Kurakin Alexey, Goodfellow Ian J., and Bengio Samy. Adversarial Machine Learning at Scale // ArXiv. — 2016. — Vol. abs/1611.01236. — Access mode: https://api.semanticscholar.org/CorpusID:9059612.
132. Qin Chongli, Martens James, Gowal Sven, Krishnan Dilip, Dvijotham Krishnamurthy, Fawzi Alhussein, De Soham, Stanforth Robert, and Kohli Pushmeet. Adversarial Robustness through Local Linearization // Advances in Neural Information Processing Systems / ed. by Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., and Garnett R. — Curran Associates, Inc. — 2019. — Vol. 32. — Access mode: https://proceedings.neurips.cc/paper_files/paper/2019/file/0defd533d51ed0a10c5c9dbf93ee78a5-Paper.pdf.
133. Wang Yisen, Zou Difan, Yi Jinfeng, Bailey James, Ma Xingjun, and Gu Quanquan. Improving Adversarial Robustness Requires Revisiting Mis-

classified Examples // International Conference on Learning Representations. — 2020. — Access mode: https://api.semanticscholar.org/CorpusID:211548864.

134. Kannan Harini, Kurakin Alexey, and Goodfellow Ian J. Adversarial Logit Pairing // ArXiv. — 2018. — Vol. abs/1803.06373. — Access mode: https://api.semanticscholar.org/CorpusID:3973828.
135. Engstrom Logan, Ilyas Andrew, and Athalye Anish. Evaluating and understanding the robustness of adversarial logit pairing // arXiv preprint arXiv:1807.10272. — 2018.
136. Mao Chengzhi, Zhong Ziyuan, Yang Junfeng, Vondrick Carl, and Ray Baishakhi. Metric Learning for Adversarial Robustness // Advances in Neural Information Processing Systems / ed. by Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., and Garnett R. — Curran Associates, Inc. — 2019. — Vol. 32. — Access mode: https://proceedings.neurips.cc/paper_files/paper/2019/file/c24cd76e1ce41366a4bbe8a49b02a028-Paper.pdf.
137. Zhang Jingfeng, Xu Xilie, Han Bo, Niu Gang, zhen Cui Li, Sugiyama Masashi, and Kankanhalli Mohan S. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger // International Conference on Machine Learning. — 2020. — Access mode: https://api.semanticscholar.org/CorpusID:211506760.
138. Cai Qi-Zhi, Du Min, Liu Chang, and Song Dawn Xiaodong. Curriculum Adversarial Training // ArXiv. — 2018. — Vol. abs/1805.04807. — Access mode: https://api.semanticscholar.org/CorpusID:44141172.
139. Wang Yisen, Ma Xingjun, Bailey James, Yi Jinfeng, Zhou Bowen, and Gu Quanquan. On the Convergence and Robustness of Adversarial Training // Proceedings of the 36th International Conference on Machine Learning / ed. by Chaudhuri Kamalika and Salakhutdinov Ruslan. — PMLR. — 2019. — 09–15 Jun. — Vol. 97 of Proceedings of Machine Learning Re-

search. — P. 6586–6595. — Access mode: https://proceedings.mlr.press/v97/wang19i.html.

140. Dauphin Yann, Pascanu Razvan, Çaglar Gülçehre, Cho Kyunghyun, Ganguli Surya, and Bengio Yoshua. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization // ArXiv. — 2014. — Vol. abs/1406.2572. — Access mode: https://api.semanticscholar.org/CorpusID:11657534.
141. Tramèr Florian, Papernot Nicolas, Goodfellow Ian J., Boneh Dan, and Mcdaniel Patrick. The Space of Transferable Adversarial Examples // ArXiv. — 2017. — Vol. abs/1704.03453. — Access mode: https://api.semanticscholar.org/CorpusID:15309391.
142. Balaji Yogesh, Goldstein Tom, and Hoffman Judy. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets // CoRR. — 2019. — Vol. abs/1910.08051. — arXiv : 1910.08051.
143. Ding Gavin Weiguang, Sharma Yash, Lui Kry Yik-Chau, and Huang Ruitong. MMA Training: Direct Input Space Margin Maximization through Adversarial Training // International Conference on Learning Representations. — 2018. — Access mode: https://api.semanticscholar.org/CorpusID:54447167.
144. Cheng Minhao, Lei Qi, Chen Pin-Yu, Dhillon Inderjit S., and Hsieh Cho-Jui. CAT: Customized Adversarial Training for Improved Robustness // ArXiv. — 2020. — Vol. abs/2002.06789. — Access mode: https://api.semanticscholar.org/CorpusID:211132607.
145. Yang Yao-Yuan, Rashtchian Cyrus, Zhang Hongyang, Salakhutdinov Ruslan, and Chaudhuri Kamalika. A Closer Look at Accuracy vs. Robustness // arXiv: Learning. — 2020. — Access mode: https://api.semanticscholar.org/CorpusID:220496204.
146. Schmidt Ludwig, Santurkar Shibani, Tsipras Dimitris, Talwar Kunal, and Madry Aleksander. Adversarially Robust Generalization Requires More

Data // ArXiv. — 2018. — Vol. abs/1804.11285. — Access mode: https://api.semanticscholar.org/CorpusID:13753923.

147. Uesato Jonathan, Alayrac Jean-Baptiste, Huang Po-Sen, Stanforth Robert, Fawzi Alhussein, and Kohli Pushmeet. Are Labels Required for Improving Adversarial Robustness? // Neural Information Processing Systems. — 2019. — Access mode: https://api.semanticscholar.org/CorpusID:173188378.
148. Carmon Yair, Raghunathan Aditi, Schmidt Ludwig, Duchi John C, and Liang Percy S. Unlabeled Data Improves Adversarial Robustness // Advances in Neural Information Processing Systems / ed. by Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., and Garnett R. — Curran Associates, Inc. — 2019. — Vol. 32. — Access mode: https://proceedings.neurips.cc/paper_files/paper/2019/file/32e0bd1497aa43e02a42f47d9d6515ad-Paper.pdf.
149. Zhai Runtian, Cai Tianle, He Di, Dan Chen, He Kun, Hopcroft John E., and Wang Liwei. Adversarially Robust Generalization Just Requires More Unlabeled Data // ArXiv. — 2019. — Vol. abs/1906.00555. — Access mode: https://api.semanticscholar.org/CorpusID:173990256.
150. Zhang Hongyi, Cisse Moustapha, Dauphin Yann N., and Lopez-Paz David. mixup: Beyond Empirical Risk Minimization. — 2018. — 1710.09412.
151. Lee Saehyung, Lee Hyungyu, and Yoon Sungroh. Adversarial Vertex Mixup: Toward Better Adversarially Robust Generalization // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2020. — June.
152. Nie Weili, Guo Brandon, Huang Yujia, Xiao Chaowei, Vahdat Arash, and Anandkumar Animashree. Diffusion Models for Adversarial Purification // Proceedings of the 39th International Conference on Machine Learning / ed. by Chaudhuri Kamalika, Jegelka Stefanie, Song Le, Szepesvari Csaba, Niu Gang, and Sabato Sivan. — PMLR. — 2022. — 17–23 Jul. — Vol. 162

of Proceedings of Machine Learning Research. — P. 16805–16827. — Access mode: https://proceedings.mlr.press/v162/nie22a.html.

153. Sohl-Dickstein Jascha Narain, Weiss Eric A., Maheswaranathan Niru, and Ganguli Surya. Deep Unsupervised Learning using Nonequilibrium Thermodynamics // ArXiv. — 2015. — Vol. abs/1503.03585. — Access mode: https://api.semanticscholar.org/CorpusID:14888175.
154. Ho Jonathan, Jain Ajay, and Abbeel Pieter. Denoising Diffusion Probabilistic Models // Advances in Neural Information Processing Systems / ed. by Larochelle H., Ranzato M., Hadsell R., Balcan M.F., and Lin H. — Curran Associates, Inc. — 2020. — Vol. 33. — P. 6840–6851. — Access mode: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.
155. Song Yang, Sohl-Dickstein Jascha, Kingma Diederik P, Kumar Abhishek, Ermon Stefano, and Poole Ben. Score-Based Generative Modeling through Stochastic Differential Equations // International Conference on Learning Representations. — 2020.
156. Song Yang and Ermon Stefano. Generative modeling by estimating gradients of the data distribution. — 2019. — Vol. 32.
157. Dockhorn Tim, Vahdat Arash, and Kreis Karsten. Score-Based Generative Modeling with Critically-Damped Langevin Diffusion // International Conference on Learning Representations. — 2021.
158. Jolicoeur-Martineau Alexia, Piche-Taillefer Remi, des Combes Rémi Tachet, and Mitliagkas Ioannis. Adversarial score matching and improved sampling for image generation // ArXiv. — 2021. — Vol. abs/2009.05475.
159. Jolicoeur-Martineau Alexia, Li Ke, Piché-Taillefer Rémi, Kachman Tal, and Mitliagkas Ioannis. Gotta Go Fast When Generating Data with Score-Based Models. — 2021.
160. Karras Tero, Aittala Miika, Aila Timo, and Laine Samuli. Elucidating the Design Space of Diffusion-Based Generative Models // arXiv preprint

arXiv:2206.00364. — 2022.

161. Pontryagin L.S. Mathematical Theory of Optimal Processes. Classics of Soviet Mathematics. — CRC Press, 2018. — ISBN: 9781351433075. — Access mode: https://books.google.com.ua/books?id=l3dZDwAAQBAJ.
162. Kloeden Peter E. Stochastic differential equations // Mathematical Proceedings of the Cambridge Philosophical Society. — 1955. — Vol. 51. — P. 663 – 677. — Access mode: https://api.semanticscholar.org/CorpusID:14529726.
163. Tramer Florian, Carlini Nicholas, Brendel Wieland, and Madry Aleksander. On Adaptive Attacks to Adversarial Example Defenses // Advances in Neural Information Processing Systems. — 2020. — Vol. 33.
164. Shi Changhao, Holtz Chester, and Mishne Gal. Online Adversarial Purification based on Self-Supervision // ArXiv. — 2021. — Vol. abs/2101.09387. — Access mode: https://api.semanticscholar.org/CorpusID:231698813.
165. Yoon Jongmin, Hwang Sung Ju, and Lee Juho. Adversarial Purification with Score-based Generative Models // Proceedings of the 38th International Conference on Machine Learning / ed. by Meila Marina and Zhang Tong. — PMLR. — 2021. — 18–24 Jul. — Vol. 139 of Proceedings of Machine Learning Research. — P. 12062–12072. — Access mode: https://proceedings.mlr.press/v139/yoon21a.html.
166. Li Xuechen, Wong Ting-Kam Leonard, Chen Ricky T. Q., and Duvenaud David. Scalable Gradients for Stochastic Differential Equations // Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics / ed. by Chiappa Silvia and Calandra Roberto. — PMLR. — 2020. — 26–28 Aug. — Vol. 108 of Proceedings of Machine Learning Research. — P. 3870–3882. — Access mode: https://proceedings.mlr.press/v108/li20i.html.

Додаток А

Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

Список публікацій здобувача за темою дисертації.

1. A. Ivaniuk and G. Kriukova, ”On Geometric Properties of Adversarial Examples,”2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, 2021, pp. 964-967, doi: 10.1109/IDAACS53288.2021.9660991.
2. Ivaniuk, A. 2022. Мовне моделювання аудіо з допомогою механізму уваги з рухомим середнім. Могилянський математичний журнал. 5, (Груд 2022), 53–56. DOI:https://doi.org/10.18523/2617-70805202253-56.
3. A. Ivaniuk (2024). ”Latent diffusion model for speech signal processing.”Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems, vol. 61, pp. 43-51, 2024. https://doi.org/10.26565/2304-6201-2024-62-05

А.1. Відомості про апробацію результатів дисертації

Основні результати дослідження доповідалися на наукових конференціях різного рівня. Це такі конференції:

- Десята всеукраїнська наукова конференція молодих математиків, Київ, 16–17 квітня 2021 р., онлайн, секційна доповідь;
- The 11th IEEE International conference on Intelligent Data Acquisition

and Advanced computing systems: Technology and Applications, 22-25 вересня, 2021 р, онлайн, секційна доповідь;

- 14 Українська конференція алгебри, Сумський державний педагогічний університет ім. А.С. Макаренка, КНУ ім. Тараса Шевченка, 3-7 липня 2023 р, онлайн, секційна доповідь;

Документ підписано у сервісі Вчасно (продовження)
Dissertation_text_Ivaniuk_final_compressed.pdf

Документ відправлено: 19:33 18.09.2024

Власник документу

Електронний підпис
19:33 18.09.2024
Ідентифікаційний код: 3554413113
ІВАНЮК АНДРІЙ ОЛЕГОВИЧ
Власник ключа: ІВАНЮК АНДРІЙ ОЛЕГОВИЧ
Час перевірки КЕП/ЕЦП: 19:33 18.09.2024
Статус перевірки сертифікату: Сертифікат діє
Серійний номер: 5E984D526F82F38F04000000D1367001079A5A05
Тип підпису: удосконалений