МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра дискретной математики и алгоритмики

ВОЙНОВ Дмитрий Михайлович

ПРОБЛЕМА УСТОЙЧИВОСТИ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ НА ПРИМЕРЕ ЗАДАЧ АНАЛИЗА БИОМЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Магистерская диссертация

специальность 1-31 80 09 "Прикладная математика и информатика"

Научный руководитель
Ковалев Василий Алексеевич
кандидат технических наук, доцент

Допущена к защите

«____» _____ 2021 г.

Зав. кафедрой дискретной математики и алгоритмики

_____ В.М. Котов

доктор физико-математических наук, профессор

Минск, 2021

ОГЛАВЛЕНИЕ

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Магистерская диссертация, 50 страниц, 10 рисунков, 1 таблица, 14 источников.

Ключевые слова: ГЛУБОКИЕ НЕЙРОННЫЕ СЕТИ, СОСТЯЗАТЕЛЬНЫЕ АТАКИ, АЛГОРИТМЫ ГЕНЕРАЦИИ АТАКУЮЩИХ ИЗОБРАЖЕНИЙ, АНАЛИЗ БИОМЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ.

Объектом исследования являются глубокие классификационные нейронные сети.

Предметом исследования является устойчивость глубоких нейронных сетей при их применении в задачах анализа биомедицинских изображений.

Целью работы было постановлено исследовать влияние состязательных атак на нейронные сети при решении различных задач классификации биомедицинских изображений; определить зависимость эффективности проводимых атак от режима их проведения, выбранного алгоритма генерации атакующих изображений, значений контрольных параметров.

В ходе работы была разработана методика экспериментального исследования, показывающего необходимые характеристики состязательных атак. В результате проведения этого исследования были определены искомые зависимости и построены соответствующие графики.

Полученный результат можно использовать в разработке безопасных систем автоматического и полуавтоматического диагностирования заболеваний, основанных на анализе изображений.

АГУЛЬНАЯ ХАРАХТАРЫСТЫКА ПРАЦЫ

Магістарская дысертацыя, 50 старонак, 10 малюнкаў, 1 табліца, 14 матэрыялаў.

Ключавыя словы: ГЛЫБОКІЯ НЕЙРОНАВЫЯ СЕТКІ, СПАБОРНЫЯ АТАКІ, АЛГАРЫТМЫ ГЕНЕРАЦЫІ АТАКУЮЧЫХ ВІДАРЫСАЎ, АНАЛІЗ БІЯМЕДЫЦЫНСКІХ ВІДАРЫСАЎ.

Аб'ектам даследавання з'яўляюцца глыбокія класіфікацыйныя нейронавыя сеткі.

Прадметам даследавання з'яўляецца ўстойлівасць глыбокіх нейронавых сетак пры іх ужыванні ў задачах аналізу біямедыцынскіх відарысаў.

Мэтай работы было пастаноўлена даследаваць уплыў спаборных атак на нейронавыя сеткі пры вырашэнні розных задач класіфікацыі біямедыцынскіх відарысаў; вызначыць залежнасць эфектыўнасці атак ад рэжыму іх правядзення, абранага алгарытму генерацыі атакуючых відарысаў, значэнняў кантрольных параметраў.

У ходзе работы была распрацавана методыка эксперыментальнага даследавання, якое паказвае неабходныя характарыстыкі спаборных атак. У выніку правядзення гэтага даследавання былі вызначаны шуканыя залежнасці і пабудаваны адпаведныя графікі.

Атрыманы вынік можна выкарыстоўваць у распрацоўцы бяспечных сістэм аўтаматычнага і паўаўтаматычнага дыягнаставання захворванняў, заснаваных на аналізе відарысаў.

ABSTRACT

Master thesis, 50 pages, 10 figures, 1 table., 14 resources.

Keywords: DEEP NEURAL NETWORKS, ADVERSARIAL ATTACKS, ALGORITHMS OF GENERATING ADVERSARIAL EXAMPLES, BIOMEDICAL IMAGE ANALYSIS.

The object of research is deep neural networks.

The subject of study is robustness of deep neural networks in the biomedical image analysis domain.

The aim of this work is to investigate the influence of adversarial attacks on neural networks in scope of different biomedical image classification problems, and to determine dependence of attacks effectiveness on its mode, algorithm of generating adversarial examples, values of control parameters.

The methodology of experimental research showing necessary characteristics of adversarial attacks was developed during the study. As a result of this research, the desired dependencies were determined and related plots were made.

The result can be applied to development of secure and safe computerized systems of automatic or semi-automatic disease diagnostics based on image analysis.

ВВЕДЕНИЕ

Одной из активно развивающихся и повсеместно применяемых тенденций современного машинного обучения является глубокое обучение – использование глубоких нейронных сетей в качестве обучаемого алгоритма. Глубокие нейронные сети показывают потрясающие результаты в огромном спектре задач машинного обучения: анализ изображений (классификация, сегментация, выделение объектов), анализ текста (определение содержания, выделение смысла), обработка звука (выделение речи). Причинами такого успеха можно назвать их способность к выявлению сложнейших зависимостей в данных, а также колоссальную «вместимость», что позволяет как ученым, так и инженерам не задумываясь выбирать нейронные сети если размер данных велик.

На сегодняшний день большинство разрабатываемых техник и исследований направлены на достижение максимальной точности работы обучаемых моделей [1]. Однако такой подход привел сообщество к страшной проблеме. Обнаружилось, что глубокие нейронные сети чрезвычайно неустойчивы: найдены специальные алгоритмы, которые минимально изменяют входное изображение, а оно, в свою очередь, по необъяснимым причинам неправильно распознается сетью [2]. При этом изменения изображения настолько малы, что зачастую неразличимы человеческим глазом. Процесс, при котором такое изображение генерируется и подается на вход сети называется состязательной атакой. Разумеется, такая проблема создает огромную брешь в безопасности нейронных сетей и ставит под сомнение целесообразность их использования в задачах с высокой ответственностью. Поэтому изучение и устранение такого эффекта является невероятно важной задачей.

В данной работе исследуется такая область применения глубоких нейронных сетей как анализ биомедицинских изображений. Использование машинного обучения в этой отрасли необходимо для решения множества задач [3, 4], решение которых становится основой для разработки так называемых систем автоматического диагностирования. На последних лежит колоссальная ответственность, так как от их работы может зависеть человеческая жизнь. Кроме того, как и в любой другой предметной области в ней, помимо общих задач и характеристик, имеется своя специфика. Поэтому проведение исследования на примере задач из данной области раскрывает не только общую проблему, но и дает больше знаний о непосредственно предметной области.

Поскольку научное сообщество все еще не видит полной картины происходящего и не имеет полностью обоснованного и фундаментального представления о происхождении такого явления, весьма ценными являются научные работы направленные, как на разработку методов его устранения, так и на разностороннее его изучение. Данная работа заключается в основном в эмпирическом исследовании проблемы и ориентирована на проведение многочисленных экспериментов, освещающих эффект состязательных атак различных типов в области анализа биомедицинских изображений. Такой метод исследования является весьма важным, поскольку, во-первых, таким образом возможно обнаружение различного рода закономерностей, существующих в проблеме, во-вторых, наработка эмпирического материала является неотъемлемой частью получения любого синтетического человеческого знания.

ГЛАВА 1
НЕЙРОННЫЕ СЕТИ

В данной работе изучается проблема безопасности глубоких нейронных сетей. Под глубокими нейронными сетями обычно подразумевают нейронные сети, которые построены из большого количества составных компонент. Помимо значительного количественного отличия, современные архитектуры нейронных сетей имеют определенные качественные особенности. В результате, методы обучения, техника работы и области применения также качественно отличаются. Рассмотрим ключевые аспекты этого явления для понимания далее изложенного материала.

1.1 Полносвязные нейронные сети

Для начала рассмотрим полносвязные нейронные сети как тип нейронных сетей, который одним из первых применялся в задачах анализа изображений.

Атомарным элементом вычислений в полносвязных нейронных сетях является *нейрон*. Структурно он состоит из следующих частей:

- Несколько входов, принимающих некоторые числовые входящие сигналы x_i
- Вычислительный центр, агрегирующий эти сигналы с использованием числовых “весов” (коэффициентов) w_i – изменяемых параметров
- Несколько выходов, распространяющих полученный числовой сигнал y

В качестве функции, агрегирующей входящие сигналы, обычно берут взвешенную сумму:

$$y = w_0 + \sum_{i=1}^{n} w_i x_i \quad (1)$$

Источником входящих сигналов могут быть исходные данные, а также выходы других нейронов. Стоит отметить, что таким образом у одного нейрона имеется ровно $n + 1$ обучаемых параметров, где n – количество входных сигналов.

Вышеописанные нейроны объединяются в так называемые *слои*. Один такой слой состоит из нескольких параллельных нейронов, которые, разделяют общие входящие сигналы и общих приемников их выходящих сигналов (рисунок 1.1).

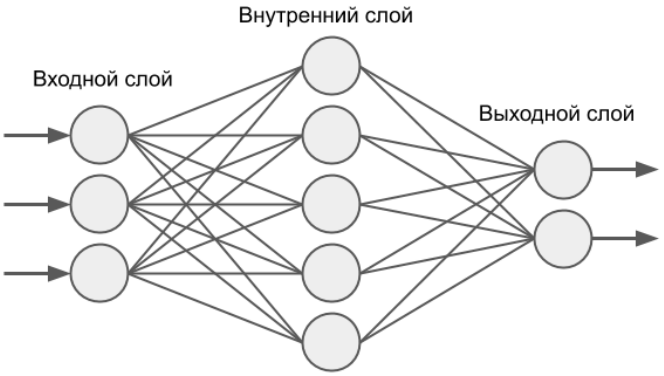


Рисунок 1.1 – Схема полносвязной нейронной сети с одним внутренним (скрытым) слоем

Кроме таких слоев отдельно выделяют входной слой, который рассматривается исключительно как распространитель входных данных следующему слою и не совершает никаких вычислений, и выходной слой, размерность которого определяет размерность выходной функции. Стоит отметить, что таким образом в слое с m нейронами и входными данными размерностью n всего $m(n + 1)$ весов.

Соединяя, в соответствии с вышеописанным, несколько слоев с разным (или одинаковым) количеством нейронов друг с другом, получим полносвязную

нейронную сеть. Она, как результат, является функцией, принимающей на вход вектор чисел, размера входного слоя и возвращающей вектор чисел, размера выходного слоя.

Однако при такой конфигурации вся нейронная сеть суть линейная функция, поскольку является композицией линейных преобразований. Для введения нелинейности в работу нейронной сети были разработаны *функции активации.* Они модифицируют нейрон так, что его выход выглядит следующим образом:

$$y = f(w_0 + \sum_{i=1}^{n} w_i x_i) \quad (2)$$

Где $f(x)$ – функция активации. Также функции активации иногда рассматривают отдельно от непосредственно полносвязного слоя и называют *слоем активации*. Исторически до сегодняшнего дня использовались многие функции активации, приведем некоторые из них:

- Функция ReLU (англ. Rectifier Linear Unit)
- Сигмоидная функция
- Гиперболический тангенс
- Арктангенс

На рисунке 1.2 изображены графики приведенных функций вблизи нуля. Кроме приведенных функций также существуют и многие другие, но они реже используются. Заметим, что эти функции не имеют никаких изменяемых параметров, однако исследователи рассматривают и изучают функции активации, содержащие обучаемые параметры. Кроме того, стоит отметить, что эти функции всюду дифференцируемы, за исключением ReLU, производную которой при необходимости просто определяют в нуле.

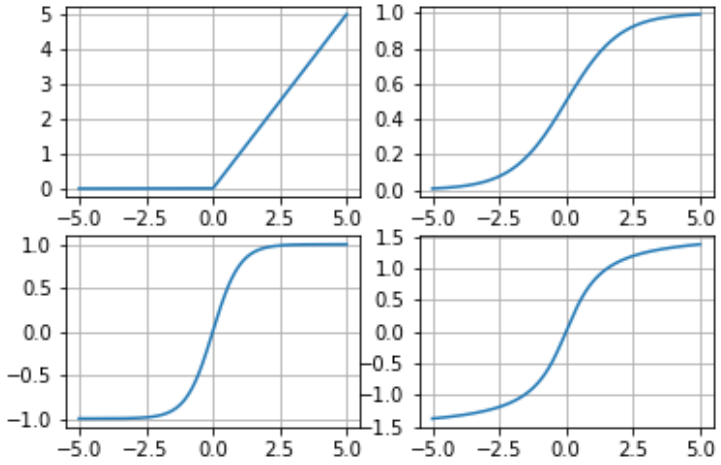


Рисунок 1.2 – Графики приведенных функций активации. Слева направо и сверху вниз: ReLU, сигмоида, гиперболический тангенс и гиперболический арктангенс

1.2 Понятие сверточного слоя нейронной сети

Исследователями был обнаружен один существенный недостаток полносвязных нейронных сетей: при большом размере входных данных либо получается большое количество параметров, либо сеть недостаточно хорошо приближает целевую функцию, если в ней существуют сложные зависимости. Это существенно усложнило использование таких сетей в анализе изображений, поскольку либо приходилось использовать изображения маленького разрешения, либо предварительно вычислять дескриптивные характеристики изображения, которые не всегда оказывались достаточно эффективными в решении поставленной задачи.

Для решения этой проблемы были разработаны *сверточные нейронные сети*. Атомарным элементом вычислений является применение корреляционного фильтра. В научном сообществе эту операцию в контексте нейронных сетей называют *сверткой*. Стоит отметить, что, вообще говоря,

операции свертки и применения корреляционного фильтра это две разные операции, однако в текущем контексте все же используется термин *свертка.*

Вычисление свертки использует так называемое *ядро свертки* (*сверточный фильтр)*. Ядро свертки представляет собой небольшую квадратную числовую матрицу обучаемых параметров, обычно нечетного размера. Свертка работает по следующему принципу:

- Входные данные представляются в виде матрицы
- Применяя операцию свертки с заданным ядром методом скользящего окна матрица преобразуется в новую матрицу меньшего размера.

При помощи таких операций свертки формируют *сверточные слои*. Для этого определяют одно или несколько ядер свертки (обычно одного размера), а вышеописанную последовательность действий совершают для каждого ядра отдельно и на выходе получается несколько уменьшенных матриц. Аналогично с полносвязными нейронными сетями здесь также используют функции активации, которые применяют поэлементно к полученному выходу.

В результате операции свертки для сверточного ядра размером $(2p + 1)\times(2p + 1)$ с матрицей весов W, с матрицей свободных членов W_0 и с входной матрицей X размера $n\times n$ получается матрица Y размера $(n - p)\times(n - p)$, элементы которой вычисляются следующим образом:

$$Y_{kl} = f\left(W * X_{[k:\, k+2p][l:\, l+2p]} + W_0\right) \qquad (3)$$

где под «*» понимается поэлементное умножение матриц. Стоит отметить, что количество обучаемых параметров сверточного слоя с m ядрами размера $p\times p$ равно mp^2. Так как обычно сверточные фильтры делают небольших размеров (3x3 – 11x11), то в итоге получается относительно небольшое количество обучаемых параметров.

1.3 Специальные слои в нейронных сетях

Кроме сверточных слоев в таких нейронных сетях используется также *слой субдискретизации* (с англ. pooling - пуллинг) – слой, принимающий на вход вектор или матрицу чисел и по некоторому правилу удаляющий определенную долю значений. Например двумерный max-pooling слой принимает на вход числовую матрицу, делит ее на непересекающиеся подматрицы размером 2x2, и из каждой подматрицы оставляет только наибольший элемент (остальные удаляет). Полученная в результате такой операции матрица имеет вдвое меньшие размерности.

Также применяется так называемый *слой пакетной нормализации* – слой, проводящий нормировку входного пакета данных. А именно, из поступающих на вход данных он поэлементно вычитает некоторое пред-обученное среднее значение, а затем делит результат на пред-обученное значение стандартного отклонения. Стоит отметить, что классическое понятие нормировки данных подразумевает сдвиг и масштабирование на значения, вычисленные по входящим данным, однако в данном случае эти параметры подбираются на этапе обучения нейронной сети.

1.4 Глубокие нейронные сети

Сверточные нейронные сети составляются из последовательных соединений сверточных слоев с различным количеством ядер свертки, слоев субдискретизации и пакетной нормализации. В результате применения такой сверточной сети к цифровому изображению получим некоторое псевдо-изображение меньшего размера. Однако в случае задачи классификации требуется иметь ответ в виде номера класса анализируемого изображения.

Ввиду этого получила распространение архитектура, которая строится из соединения сверточной нейронной сети и полносвязной. В результате сеть работает следующим образом:

- Сверточная нейронная сеть выделяет из входных данных так называемый вектор признаков (вектор получается из псевдо-изображения путем его “вытягивания” в одномерную последовательность чисел). По этой причине принято называть сверточную часть нейронной сети экстрактором признаков. Полученные признаки имеют меньший размер, чем сами входные данные.
- Полносвязная нейронная сеть принимает на вход полученный вектор и выдает на выход номер класса (или другую числовую конфигурацию, из которой при помощи элементарных преобразований этот номер класса можно получить). Полносвязную часть принято называть классификатором.

Современные классификационные нейронные сети основываются на этом принципе, так как он сочетает в себе наличие относительно небольшого количества обучаемых параметров, что позволяет нейронной сети занимать меньше памяти, и высокое качество работы.

Вместе с развитием идей нейронных сетей развивались вычислительные мощности компьютеров. Активно стали распространяться графические карты, которые позволяли проводить все больше и больше параллельных вычислений за то же время, что благоприятно влияло на развитие нейросетевых алгоритмов. Появилась возможность создавать нейронные сети все большего и большего размера, которые при этом обучались и работали за приемлемое время. В итоге научное сообщество начало массово изучать и использовать глубокие нейронные сети, экспериментируя с архитектурами, создавая новые агрегации слоев различных видов (рисунок 1.3).

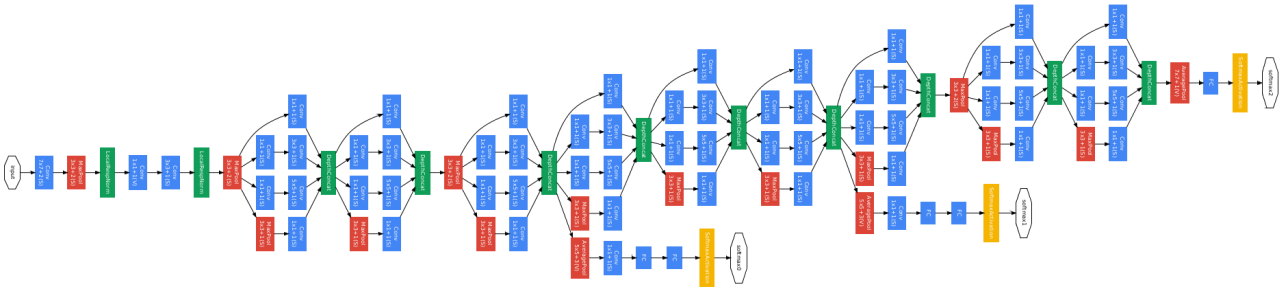


Рисунок 1.3 – Схематическое изображение архитектуры нейронной сети InceptionV3. Как видно в ней использовано почти 100 слоев различного типа, причем некоторые из них соединены параллельно

В чем принципиальное отличие нейронных сетей с большим количеством слоев от небольших сетей? Как было найдено, глубокие нейронные сети способны решать сложнейшие задачи классификации, обучаясь на миллионах изображений. Во-первых, они показывают огромную «вместимость» данными, во-вторых способность выявлять сложные распределения в них и, несмотря на огромное количество обучаемых параметров, которое достигает десятков миллионов значений, высокую скорость работы при использовании графических карт. Поэтому их постоянно используют в задачах машинного обучения, в которых имеется большое количество данных. Однако, как будет показано далее, вместе с огромным успехом такие нейронные сети принесли проблемы безопасности и устойчивости работы. Так как сложные системы, основанные на машинном обучении, не смогут существовать без глубоких нейронных сетей, то перед нами стоит задача изучения и решения возникших проблем.

ГЛАВА 2
СОСТЯЗАТЕЛЬНЫЕ АТАКИ

2.1 Понятие состязательных атак

Под состязательной атакой понимается процесс, в результате которого некоторый атакуемый классификатор неестественно предсказывает определенное изображение неверно с точки зрения человека. Под неестественностью понимается тот факт, что изображения, которые классификатор предсказывает неправильно, являются на первый взгляд нормальными допустимыми изображениями соответствующей предметной области. При этом отличие таких ошибок от простых ошибок обобщения в том, что для данных изображений обычно есть практически не отличающееся от него парное, при этом правильно распознаваемое сетью.

В данной работе само исследование состязательных атак будет проводиться на примере задачи классификации биомедицинских изображений. Однако стоит отметить, что атаки такого рода можно проводить и в рамках других задач машинного обучения (например, анализ звука).

Дадим небольшой обзор существующих способов проведения состязательных атак в задачах анализа изображений.

2.2 Генерация атакующих изображений

Первым и по сей день основным способом проведения состязательных атак является генерация атакующих изображений – искусственных изображений, слабо отличающихся от «натуральных» из определенной выборки данных. Атакующие изображения генерируются с помощью специальных алгоритмов. Благодаря построению таких алгоритмов и некоторым другим

факторам (о них позже) подача таких изображений на вход нейронной сети зачастую заканчивается ошибкой предсказания.

Генерация атакующих изображений заключается в последовательном выполнении следующих трех шагов: (1) выбор изображения из предметной области атакуемой нейронной сети; (2) генерация особого шумоподобного возмущения при помощи специального алгоритма; (3) применение полученного возмущения к выбранному изображению путем обычного поканального сложения. В результате получим искомое атакующее изображение, которое, вероятно, будет ошибочно предсказано классификационной сетью.

На первый взгляд состязательные атаки такого рода могут показаться несущественной проблемой, поскольку для их проведения требуется каким-то образом подавать атакующие изображения прямо на вход работающей сети, которая так или иначе является частью некоторого ПО, установленного на вычислительный аппарат. Последнее подразумевает тот факт, что вход нейронной сети доступен только некоторому физическому устройству, получающему информацию из реального мира и, как следствие, скрыт от внешних глаз, что делает получение доступа к ней такого рода еще одной трудоемкой задачей. Но ничего из этого не гарантирует, что атакующие изображения могут быть лишь результатом работы алгоритмов. До сих пор мало что известно о любого рода естественном способе обнаружения атакующих изображений, но разнообразие их искусственных представителей заставляет сообщество детально разбираться в вопросе. Кроме того, изучение этого явления расширяет знания сообщества о многих аспектах работы глубоких нейронных сетей, которые остаются нераскрытыми.

2.3 Состязательные атаки из реального мира

Существуют способы проводить атаки таким образом, что атакующий агент подается не напрямую на вход сети, а на вход физического устройства. Притом, если в случае с атакующими изображениями подаются одиночные изображения, то тут речь идет о непрерывной череде ошибок распознавания объектов при видеосъемке. В работе [5] приведен пример таких состязательных атак. Идея заключается в том, чтобы заранее сгенерировать некоторый вредоносный узор, затем распечатать его и показать камере, в ПО которой включена функция распознавания объектов (которая, обычно, основана на глубоких нейронных сетях, например архитектуры YOLO). В результате устройство не распознает ни распечатанный паттерн, ни человека, находящегося за ним. Несмотря на кардинальное отличие этого способа атаки от предыдущего, он тесно связан с идеей генерации атакующих изображений. Поэтому последняя является важной ветвью проблемы.

Данная работа посвящена изучению состязательных атак, основанных на генерации атакующих изображений различного типа. Приведем формальное определение, рассмотрим и классифицируем существующие алгоритмы генерации.

ГЛАВА 3
АТАКУЮЩИЕ ИЗОБРАЖЕНИЯ

3.1 Определение атакующего изображения

Пусть $x \in R^d$ – нормализованное входное изображение, $y: R^d \to (0,1)^p$ – выход классификационной нейронной сети как функции от входного изображения с количеством классов p, $F: (0,1)^p \to \{1, ..., p\}$ – решающая функция классификации (в данной работе рассматривается функция argmax) . Пусть $\epsilon > 0$ – некоторое небольшое положительное число. Тогда ϵ – атакующим изображением называется такое изображение , что верно:

$$F\left(y\left(x^*\right)\right) \neq F\left(y\left(x\right)\right) \tag{4}$$

При выполнении:

$$||x^* - x|| \leq \epsilon \tag{5}$$

В последнем уравнении в качестве нормы рассматривают обычно L_2 или L_∞, но, разумеется, допустима любая норма.

Уравнение (4) показывает то, что результат классификации атакующего изображения отличен от такового у исходного изображения. Но это обычная ситуация, например для изображений из другого класса. Для того, чтобы показать неестественность эффекта вводится ограничение (5) для небольшого ϵ. Приемлемое значение параметра (такое, чтобы атакующее изображение было достаточно близким к исходному изображению) выбирается атакующим лицом. Обычно для того, чтобы заставить классификатор ошибиться, достаточно

выбрать небольшое ϵ. Этот параметр называется магнитудой модификации, поскольку условие (5) можно переписать в виде:

$$\|\Delta x\| \leq \epsilon, x = x + \Delta x \quad (6)$$

В таком случае ϵ является магнитудой модификации изображения x. Разумеется, при разных значениях ϵ модификация изображения в различной степени заметна (рисунок 3.1). Многие работы показывают, что часто можно изменить изображение незаметным для глаза человека образом и все равно заставить классификатор ошибиться.

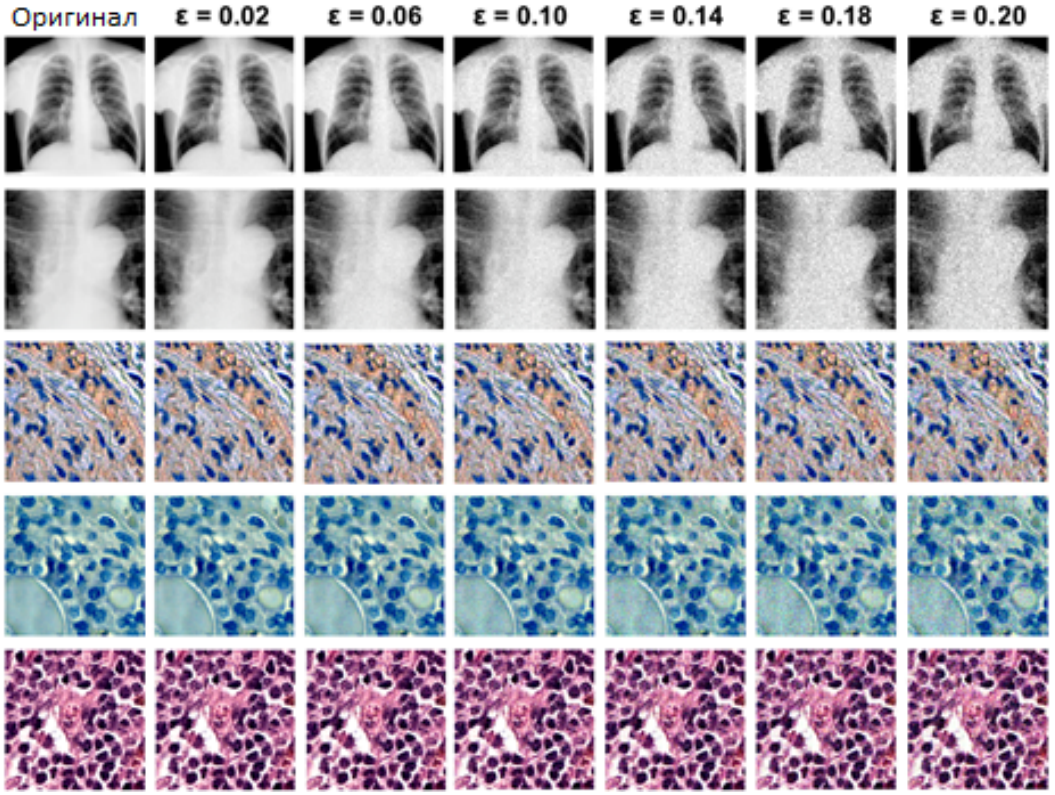


Рисунок 3.1 – Примеры атакующих изображений, сгенерированных с возрастающей допустимой амплитудой для некоторых из используемых наборов данных.

Как сказано ранее, атакующие изображения являются результатом работы специальных алгоритмов генерации (обнаружения). Они будут рассмотрены в последующих главах.

3.2 Известные свойства атакующих изображений

Несмотря на то, что не существует полного и точного представления об истинном происхождении атакующих изображений, исследователями уже обнаружены некоторые воспроизводимые их свойства [6, 7, 8, 9]:

- Хоть в теории небольшие изменения входного изображения не должны сильно влиять на результат работы сети, на практике оказывается, что для проведения успешной атаки часто достаточно атакующих изображений с минимальной модификацией.
- Атакующим изображениям характерна переносимость: изображение, сгенерированное для атаки определенной сети, можно использовать для атаки другой сети, которая предназначена для классификации изображений того же типа, что и первая сеть. Оказывается, что часто и такая атака успешна.
- Разница выходов сверточной части нейронной сети для оригинального и атакующего изображений достаточно велика и может быть больше, чем разница между двумя оригинальными изображениями разных классов.

ГЛАВА 4
АЛГОРИТМЫ ГЕНЕРАЦИИ АТАКУЮЩИХ ИЗОБРАЖЕНИЙ

На сегодняшний день разработано достаточно много алгоритмов генерации атакующих изображений. Среди них выделяются как общие концепции, так и множество так называемых эвристик. Для начала рассмотрим общую идею существующих алгоритмов, а затем приведем несколько конкретных из них.

4.1 Классификация алгоритмов генерации атакующих изображений

По информации, требуемой для работы, алгоритмы генерации делят на атаки по методу белого и черного ящика.

Для проведения атаки по методу белого ящика необходимо владеть информацией о конфигурации сети: архитектуру и обученные параметры. Также требуется иметь оригинальное изображение соответствующей предметной области для генерации самого атакующего изображения.

Для проведения атаки по методу черного ящика достаточно иметь доступ к подаче изображений на вход сети, доступ к результатам предсказания, а конфигурация сети остается неизвестной. Разумеется также необходима информация о предметной области распознаваемых изображений.

По предсказанному классу атакующего изображения алгоритмы генерации делят на направленные и ненаправленные.

Метод проведения направленных атак позволяет заранее определить класс атакующего изображения предсказанный сетью. Ненаправленные атаки, в свою очередь, стремятся лишь обмануть сеть, без гарантий на то какой именно класс получится в результате предсказания. Большинство алгоритмов генерации

атакующих изображений могут быть сконфигурированы как оба типа. В данной работе рассмотрены исключительно ненаправленные атаки.

4.2 Основная концепция алгоритмов генерации атакующих изображений

Базовую идею алгоритмов генерации атакующих изображений можно вывести из идей алгоритмов обучения нейронной сети. Опишем ее.

Нейронная сеть по своей сути является некоторой функцией от входного изображения x и изменяемых весов θ. Задача обучения состоит в том, чтобы минимизировать функцию потерь $L(f(x, \theta), S)$, где $S = \left\{\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)\right\}$ – тренировочная выборка. Переменной в данной задаче являются веса θ. Учитывая то, что функция имеет определенный вид в зависимости от архитектуры и известна обучающему алгоритму, существует множество математических способов решить эту задачу в какой-то степени.

Если в вышеописанной задаче вместо минимизации функции потерь проводить максимизацию, вместо тренировочной выборки взять одну пару изображение-метка $\{(x^{(k)}, y^{(k)})\}$ для некоторого k, зафиксировать параметр θ, в качестве переменной взять x и добавить ограничение (5), то получится задача генерации атакующего изображения для объекта $x^{(k)}$. Действительно, в такой трактовке задача состоит в поиске x^*, удовлетворяющего ограничению (5) такого, чтобы некоторая заданная сеть предсказывала x^* максимально отлично от $x^{(k)}$. Если в качестве $L(y', y)$ взять 0-1 функцию потерь, то решение такой задачи (если оно существует) совпадает с определением из главы 3.

В нашем случае нейронная сеть является непрерывной и почти всюду дифференцируемой многомерной функцией $f(x, \theta)$, поэтому, дополнив

недифференцируемые точки каким-либо приемлемым значением производной, мы имеем право вычислять частные производные и, как следствие, градиент композиции $L(f(x, \theta), S)$ (в случае дифференцируемости L). В результате можем применять различные алгоритмы оптимизации (например, градиентный спуск).

4.3 Алгоритм PGD

Данный метод есть не что иное, как применение классической техники градиентного спуска с учетом ограниченности возмущения [8]**.** В качестве функции потерь $L(y', y)$ можно рассмотреть k-тую компоненту вектора вероятностей y'. Тогда минимизация такой функции будет приводить к понижению вероятности принадлежности атакующего изображения этому классу, а максимизация к повышению.

В результате генерация направленного на класс t атакующего изображения этого метода зависит от коэффициента обучения $\alpha > 0$, количества итераций $n \in N$, магнитуды возмущений ϵ и определяется следующим образом:

$$x_{k+1} = Clip_{x,\epsilon}\left(x_k + \alpha * \nabla y_t\left(x_k\right)\right) \quad (7)$$

где $x_0 = x$, $k \in [0, n-1]$, $x^* = x_n$, а функция $Clip_{x,\epsilon}$ “обрезает” те элементы ее аргумента, которые отличаются от тех же элементов x более чем на ϵ. Генерация ненаправленного атакующего изображения этого метода зависит от исходного класса m и определяется следующим образом:

$$x_{k+1} = Clip_{x,\epsilon}\left(x_k - \alpha * \nabla y_m\left(x_k\right)\right) \quad (8)$$

Поскольку на каждой итерации применяется функция $Clip_{x,\epsilon}$, то в результате работы алгоритма получится изображение x^* автоматически удовлетворяющее ограничению (5) для L_∞ нормы.

4.4 Алгоритм Deepfool

Понятие градиента функции часто понимается как линеаризация функции. Действительно, при решении задачи приближения дифференцируемой функции некоторой линейной функцией в окрестности заданной точки в качестве вектора коэффициентов получим непосредственно градиент функции в этой точке. Возможность представить выход нейронной сети как условно линейную функцию заложила фундамент в идею следующего алгоритма.

Допустим в качестве атакуемого классификатора рассматривается линейный бинарный классификатор. Тогда процесс классификации заданного объекта суть определение относительного местоположения этого объекта и разделяющей гиперплоскости классификатора. В таком случае нетрудно догадаться, что поиск атакующего изображения заключается в поиске объекта, находящегося по другую сторону от гиперплоскости. При этом, поскольку желательно найти наиболее близкий к исходному объекту пример, в качестве атакующего объекта x выберем произвольный объект в окрестности проекции x^pобъекта на гиперплоскость классификатора.

Строго говоря для линейного классификатора порождаемого гиперплоскостью $f(x) = \langle w, x\rangle + b$ проекцию x^p заданного объекта x_0 на эту гиперплоскость можно найти по формуле:

$$x^p = x_0 - \frac{f(x_0)}{\|w\|^2} w \quad (9)$$

Такая точка должна классифицироваться как равновероятно принадлежащая обоим классам. Поскольку нас интересует непосредственно противоположная классификация, то для генерации самого атакующего изображения выберем точку немного за полуплоскостью, домножив второй член разности в формуле (9) на некоторое число $1 + \eta$.

Теперь рассмотрим многоклассовый линейный классификатор. Такой классификатор задается набором линейных функций $f_k(x) = \langle w_k, x \rangle + b_k$ и решающей функцией $F(x) = argmax_k f_k(x)$. Пусть k_0 - класс исходного объекта. Тогда условие успешности атакующего изображения x^* можно записать в виде $argmax_k f_k(x^*) \neq k_0$, или другими словами *существует класс, к которому атакующее изображение принадлежит с большей вероятностью, чем к исходному.* Последнее условие переписывается в виде $\exists l: f_l(x) - f_{k_0}(x) > 0$, что можно трактовать как отнесение к положительному классу линейным классификатором в бинарной задаче классификации с функцией гиперплоскости $f(x) = f_l(x) - f_{k_0}(x)$. Тогда для каждого класса отличного от исходного рассмотрим бинарную задачу классификации «исходный класс против текущего». Алгоритм генерации атакующего изображения для такой задачи описан выше. Теперь выберем среди найденных примеров для каждой задачи ближайший к исходному объекту и получим искомое атакующее изображение. Формула проекции для одной «псевдозадачи» классификации против класса l может быть записана как:

$$x^p = x_0 - \frac{f_l(x_0) - f_{k_0}(x_0)}{\|w_l - w_{k_0}\|^2} (w_l - w_{k_0}) \quad (10)$$

Аналогично задаче бинарной классификации для самого атакующего изображения требуется домножить второй член в разности на некоторое положительное число большее единицы.

Теперь перейдем непосредственно к нашей задаче. Обычно выход нейронной сети не является линейной функцией, поэтому для применения вышеописанного алгоритма воспользуемся линеаризацией, о которой говорилось ранее. Для начала отметим, что $y_k(x) \approx y_k\left(x_0\right) + \langle \nabla y_k\left(x_0\right), x - x_0 \rangle$. Тогда, взяв в качестве w вектор градиент выхода сети в точке, можем применить алгоритм, построенный для линейных классификаторов. По причине того, что такая линеаризация является всего лишь приближением, авторы оригинальной работы [14] предлагают применять алгоритм итеративно до тех пор, пока не найдется успешное атакующее изображение.

Несмотря на то, что данный алгоритм явно не минимизирует никакую функцию потерь и нигде не накладывает ограничений на применяемую к изображению модификацию, на практике он часто генерирует успешные атакующие изображения, которые, при этом, очень близки по L_2 норме к исходным. Также стоит отметить, что оригинальная формулировка алгоритма подразумевает проведение ненаправленных атак.

4.5 Алгоритм Карлини и Вагнера (CW)

Авторами работы [13] показано высокое качество атакующих изображений, получающихся в результате работы алгоритма L-BFGS. В 2016 году основываясь на том же подходе Карлини и Вагнер разработали более эффективный алгоритм [12]. Данный алгоритм использует подход, основанный на переформулировке задачи из 3.1. Этой задаче ставится в соответствие задача

нелинейного программирования, в которой требуется минимизировать расстояние между атакующим и исходным изображениями, при условии неверной классификации сетью. Такую задачу затем можно решать различными способами. Разумеется, точный алгоритм решения в общем случае не существует, поэтому новую задачу решают различными приблизительными алгоритмами.

Формально рассматривается следующая задача минимизации:

$$\|x^* - x\| \to min,\ F\left(y\left(x^*\right)\right) = l,\ x \in [0,1]^n \tag{11}$$

Где l- заранее выбранный класс отличный от класса исходного изображения. По сути в данной задаче необходимо найти атакующее изображение, ближайшее к исходному изображению. Таким образом любой алгоритм, который решает задачу (11), является направленным алгоритмом генерации атакующих изображений. Нетрудно видеть, что как и точное решение этой задачи, так и отдельное соблюдение условия $F\left(y\left(x^*\right)\right) = l$являются вычислительно трудоемкими задачами, поскольку выход глубокой нейронной сети представляет из себя чрезвычайно сложную функцию. По этой причине в [13] при помощи весьма распространенного подхода данное условие заменяется на дополнительный член в целевой функции, и вместо оригинальной задачи (11) рассматривается следующая:

$$||x^* - x|| + cf_l(x^*) \to min,\ x^* \in [0,\ 1]^n \tag{12}$$

где $f_l(x)$ это функция, которая по заданному изображению x возвращает некоторый показатель «отличия» от класса l: чем больше ее значение, тем с

меньшей вероятностью она относится к классу. Авторами оригинальной работы предлагается рассмотреть произвольную функцию f, которая удовлетворяет неравенству $f(x, l) \leq 0$тогда и только тогда когда $F(y(x)) = l$. Таким образом при минимизации целевой функции выгодно уменьшать функцию f и, как следствие, повышать «близость» оптимизируемой переменной к классу l. В самой работе рассматриваются несколько подходящих под эти свойства функций и на основе результатов проводимых экспериментов выбирается лучшая из них (приведена далее).

Оптимальный параметр c можно найти любым линейным оптимизационным поиском (например, двоичным поиском) – для каждого нового значения параметра задача (12) решается заново. Авторы работы предлагают брать наименьшее его значение, при котором атакующее изображение успешно.

Также недостаток в наличии ограничения $x^* \in [0, 1]^n$ исправляется заменой переменных $x^* = \frac{1}{2}(\tanh(w) + 1)$. Нетрудно видеть, что, учитывая область значений гиперболического тангенса, при любом значении w значение x^*остается в пределах $[0, 1]^n$.

В итоге ставится следующая задача оптимизации:

$$||\frac{1}{2}(\tanh(w) + 1) - x|| + cf_l(\frac{1}{2}(\tanh(w) + 1)) \to min, \text{ ν} \qquad (13)$$

где

$$f_l(x) = \max(\max_{k \neq l} y_k(x) - y_l(x), - \text{к}) \qquad (14)$$

Функция f также зависит от параметра κ показывающего желаемую степень уверенности в классификации атакующего изображения: при $\kappa = 0$ достаточно найти успешное атакующее изображение — вероятность принадлежности целевому классу которого просто больше вероятностей принадлежности другим классам, но при увеличении κ эта вероятность повышается. Далее эта функция минимизируется оптимизатором Adam и в результате после ограниченного количества итераций получаем атакующее изображение.

Отличительной чертой этого алгоритма является практически показанное качество генерируемых атакующих изображений. Получаемые изображения практически неотличимы от исходных, при этом количество успешных проводимых атак значительно больше, чем у других алгоритмов. Однако вместе с эффективностью алгоритм отличается заметно большей продолжительностью работы.

ГЛАВА 5
ИСПОЛЬЗУЕМЫЕ НАБОРЫ ДАННЫХ

5.1 Исходные наборы изображений

В проведенном исследовании использовались 5 различных наборов медицинских изображений. Приведем их краткое описание

Гистологические изображения тканей лимфоузлов, пораженных метастазами. Первоначально имелся набор полнослайдовых гистологических изображений с размерами, достигающими 100000x100000 [17]. Так как изображения такого размера целиком не подвергаются анализу, а только лишь их участки, то они были разрезаны на плитки размером 256x256. Полученные плитки были очищены от участков со стеклом или пузырями, которые не несут никакой информации. В итоге был получен набор из 125000 цветных изображений размером 256x256 представленный двумя классами: норма и участки, пораженные метастазами. Набор данных сбалансирован: изображений ровно по 62500 на класс.

Гистологические изображения тканей яичников и щитовидной железы, пораженных опухолями. Оригинальный набор данных состоял из 4000 изображений размером 2048x1536 тканей 26 пациентов с опухолями в упомянутых органах. Применив технику, аналогичную использованной в вышеописанном наборе данных был получен набор из 192000 цветных изображений размером 256x256 представленный четырьмя классами: яичники-норма, яичники-опухоль, щитовидная железа-норма, щитовидная железа-опухоль. Набор данных сбалансирован: изображений по 48000 на класс.

Рентгеновские изображения легких. Оригинальный набор данных состоял из 1908926 рентгеновских изображений грудной клетки разного размера, сделанных на разных аппаратных устройствах. Вместе с каждым изображением

имелся текстовый отчет врача радиолога, который в произвольном виде включал в себя возраст, пол и возможные заболевания, обнаруженные по снимку. При помощи специальных алгоритмов анализа текста эти данные были извлечены из отчетов, среди них выделены успешные результаты. Сами изображения подверглись нормализации по яркости и были приведены к размеру 512x512. Из полученного массива данных было выбрано подмножество из 550080 изображений мужчин и женщин в возрасте от 17 до 80 лет включительно.

Компьютерная томография легких. Первоначально набор данных состоял из множества 3D полутоновых снимков легких пораженных туберкулезом, полученных при помощи компьютерной томографии. Ввиду большой размерности трехмерных изображений они были разрезаны на двумерные “плоские” слои. Затем с целью сохранения некоторой пространственной информации каждое полученное полутоновое двумерное изображение было преобразовано в цветное путем помещения исходного изображения в зеленый канал, соседнего по слоям снизу изображения в красный канал и соседнего по слоям сверху в синий. В результате полученные цветные изображения также хранили в себе два соседних. Поскольку размерность этих изображений все еще оставалась высокой они были нарезаны на плитки размером 256x256. В результате получилось 149248 цветных изображений размером 256x256. Набор данных несбалансирован: имеются 111990 изображений со здоровыми участками и 37258 с пораженными туберкулезом.

Гистологические снимки, окрашенные шестью химическими агентами. Оригинальный набор данных состоял из полнослайдовых гистологических изображений тканей, окрашенных шестью разными химическими составами. Аналогично вышеописанным гистологическим наборам данных эти изображения также были нарезаны на плитки размером 256x256 и в результате

сформировали набор данных из 267984 цветных изображений, разделенных на 6 классов. Классовый баланс приведен в таблице 4.1.

Таким образом, исследование проводилось над большим количеством медицинских изображений различных модальностей, которые являются весьма распространенными и часто используемыми в медицине: гистологические изображения – золотой стандарт в диагностировании рака и рентгеновские изображения – основное направление в выявлении заболеваний легких, дефектов скелета.

5.2 Построенные задачи классификации

На основе пяти вышеприведенных наборов медицинских изображений были построены 8 задач классификации (таблица 5.1). Для некоторых наборов данных была отличная возможность конфигурирования нескольких задач классификации, что позволило рассмотреть задачу более детально. Как видно, в основном рассматривались задачи бинарной классификации – достаточно часто встречающиеся в вопросах диагностирования заболеваний.

Набор данных	**Сокращение**	**Задача классификации**	**Кол-во изображений**	**Кол-во изобр. по классам**	**Точность**
Гистология с метастазами	H-MT	Норма / участки с метастазами	100000	50000/50000	0.97
Гистология с опухолями в яичниках/щитовидной железе	H-OV	В яичниках: норма / опухоль	96000	48000/48000	0.92
	H-TH	В щитовидной железе: норма / опухоль	96000	48000/48000	0.94
	H-OV-TH	Яичники-норма/яичники-опухоль/щитовидная железа-норма/щит	192000	48000/48000 /48000 /48000	0.91

		овидная железа-опухоль			
Рентген норма	X-NR2	Две возрастные группы: 20-35 лет / 50-70 лет	200000	100000 / 100000	0.98
	X-NR3	Три возрастные группы: 17-24 лет / 25-41 лет / 42-80 лет	550080	183360 / 183360 / 183360	0.83
Компьютерная томография легких	CT	Норма / Туберкулез	149248	111990 / 37258	0.96
Гистологические изображения, окрашенные шестью химическими агентами	H-ST	Агенты: CD31 / CD105 / D240 / FR ES / H&E / Ki67	267984	59568 / 37488 / 55296 / 35280 / 24192 / 56160	0.95

Таблица 5.1 – Задачи классификации, построенные на основе используемых наборов данных. В последнем столбце приведены точности обученных классификационных нейронных сетей

ГЛАВА 6
ИССЛЕДОВАНИЕ АТАК БЕЛОГО ЯЩИКА

6.1 Обучение нейронных сетей

Для каждой задачи классификации из предыдущей главы была обучена нейронная сеть. В качестве архитектуры сети была выбрана сеть InceptionV3, пред-обученные веса не использовались. Для обучения сетей изображения нормировались до отрезка [0, 1]. В качестве тренировочного оптимизатора был выбран AdamOptimizer с одинаковым для всех задач обучающим коэффициентом. Во всех случаях для достижения приемлемых для исследования точностей достаточно было менее 50 эпох обучения. Все вычисления проводились на компьютере с процессором Intel® Core™ i7-6700K и двумя видеокартами Nvidia GeForce GTX 1080 Ti. В качестве библиотек для обучения нейронных сетей использовались Keras, Tensorflow.

6.2 Постановка исследования

Как было сказано ранее цель данной работы оценить влияние явления состязательных атак на глубокие нейронные сети в области анализа биомедицинских изображений. Для этого изучаются различные характеристики работы алгоритмов генерации атакующих изображений, которые применяются к нейронным сетям, обученным на вышеописанных наборах данных. В качестве основных свойств алгоритмов в данной работе рассматривались:

- Эффективность — способность найти успешное атакующее изображение при заданных ограничениях.
- Качество получаемых атакующих изображений — насколько близки к исходным изображениям получаются их атакующие версии.

Исходя из формулировки алгоритмов, приведенных в главе 4, и проведенных другими авторами опытов можно предположить, что алгоритм PGD покажет меньшую эффективность нежели DeepFool и CW при тех же размерах возмущения, что, однако, компенсируется скоростью его работы.

Теперь опишем проводимый эксперимент. Для каждой пары «задача классификации»-«алгоритм генерации» выполняется следующее:

1. На нейронную сеть, обученную для задачи классификации, для каждого изображения из тестовой выборки при помощи выбранного алгоритма проводится атака, в результате которой генерируется атакующее изображение. Для дальнейшей оценки качества алгоритмов DeepFool и CW L_2 и L_∞ нормы разности сгенерированного и исходного изображений сохраняются, а алгоритм PGD запускается для нескольких значений ϵ.
2. Атакующее изображение подается на вход этой же сети, а полученные вероятности принадлежности классам сохраняются для дальнейшего анализа.

С целью оценки эффективности работы алгоритма и качества генерируемых атакующих изображений вычисляется доля успешных атак, в результате которых норма разности атакующего и исходного изображений ограничены некоторым числом. В этой работе рассматривались L_2 и L_∞ нормы.

6.3 Зависимость успешности атак от L_∞ нормы возмущения.

Построим график зависимости доли успешных атак от L_∞ нормы применяемого возмущения (рисунок 6.1). Как было сказано выше для возможности построить такую зависимость PGD алгоритм запускается отдельно для $\epsilon = 0.02, 0.04, \ldots, 0.2$ (соответственно напрямую ограничивая L_∞

норму), а для алгоритмов DeepFool и CW L_∞ нормы возмущения вычисляются по завершению их работы.

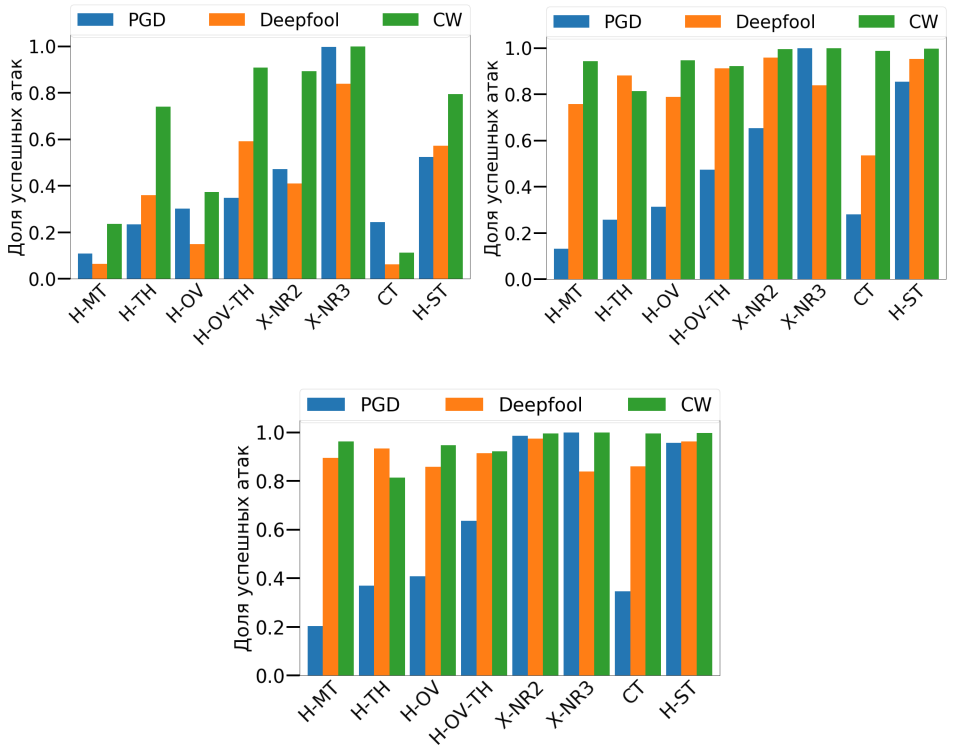


Рис.6.1 — Доля успешных атак для использованных наборов данных при L_∞ ограничении нормы возмущения в 0.02, 0.10 и 0.2 соответственно.

Как видно с точки зрения L_∞ нормы алгоритм CW показал себя лучше почти во всех случаях (кроме ϵ равного 0.10 и 0.2 на наборе данных H-TH, где алгоритм DeepFool проявил себя эффективней). Также можно отметить, что при минимальном рассмотренном эпсилон алгоритм PGD чаще немного более эффективен нежели DeepFool, однако уже при эпсилон не ниже 0.08 качество работы последнего заметно превышает PGD. В целом, как и ожидалось, алгоритм PGD показывал себя хуже, чем DeepFool и CW, несмотря на то что последние два оптимизируют L2 норму. При этом значительной эффективности хотя бы в 80% PGD не достигает в 6 из 8 случаев даже при максимальном возмущении.

6.4 Зависимость успешности атак от L_2 нормы возмущения.

Построим аналогичную зависимость, ограничивая возмущение L_2 нормой (рисунок 6.2). Данную зависимость будем строить для алгоритмов DeepFool и CW, поскольку как было показано выше PGD работает заметно хуже при средних и больших возмущениях, а запуская его только для маленького эпсилон мы не получим высокой итоговой эффективности.

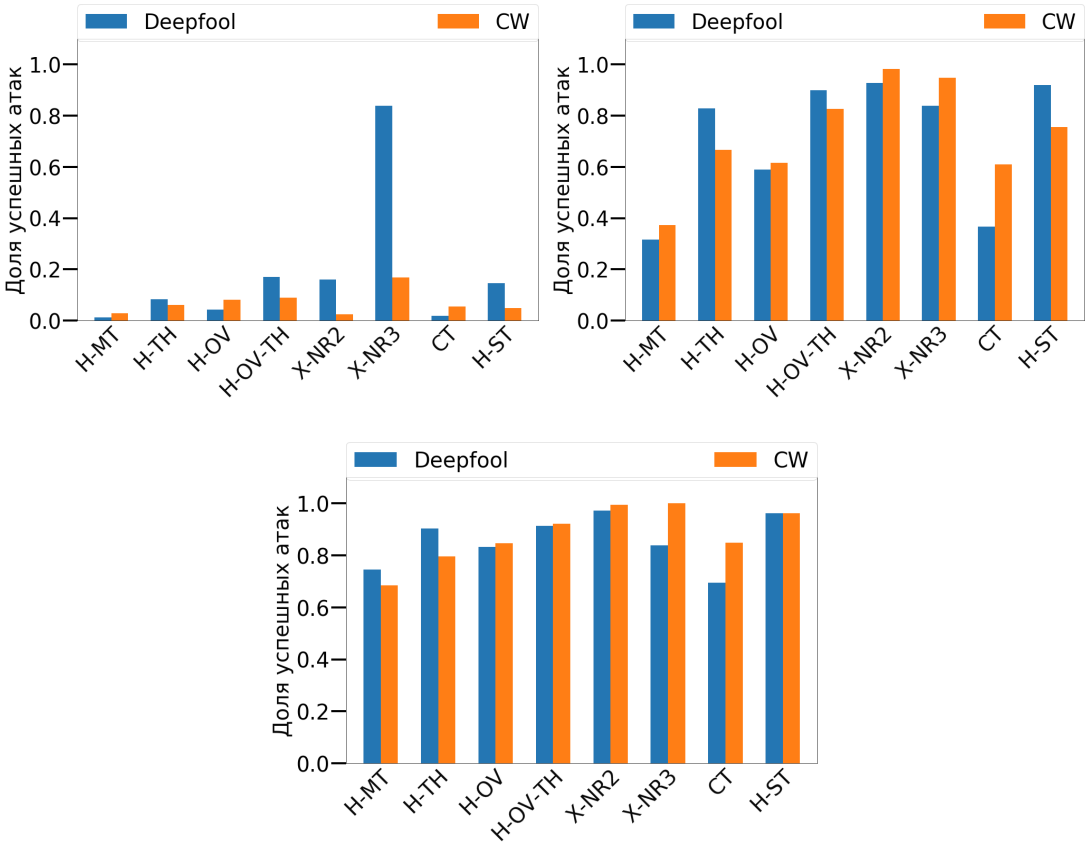


Рис. 6.2 - Доля успешных атак для восьми наборов данных при L_2 ограничении нормы возмущения в 0.1, 1.0 и 2.0 соответственно.

По графикам видно, что в целом алгоритмы DeepFool и CW близки по эффективности: для ϵ равного 1.0 и 2.0 в 4 из 8 случаев доли успешных атак обоих алгоритмов почти равны, в оставшихся случаях однозначного фаворита

нет. Стоит отметить, что для изображений размера 256x256 значение 2.0 L_2 нормы возмущения является весьма небольшой величиной: в среднем каждый пиксел меняется на 0.0078 (в условиях [0, 1] нормировки), что меньше одного процента максимально допустимого значения.

6.5 Зависимость успешности атаки от количества применяемых итераций алгоритма PGD

Для алгоритма PGD была построена зависимость успеха атаки от количества проделанных итераций алгоритма (рисунок 6.3). Исходя из построения алгоритма ожидается, что эффективность атаки будет постепенно увеличиваться, асимптотически сходясь к своему максимуму.

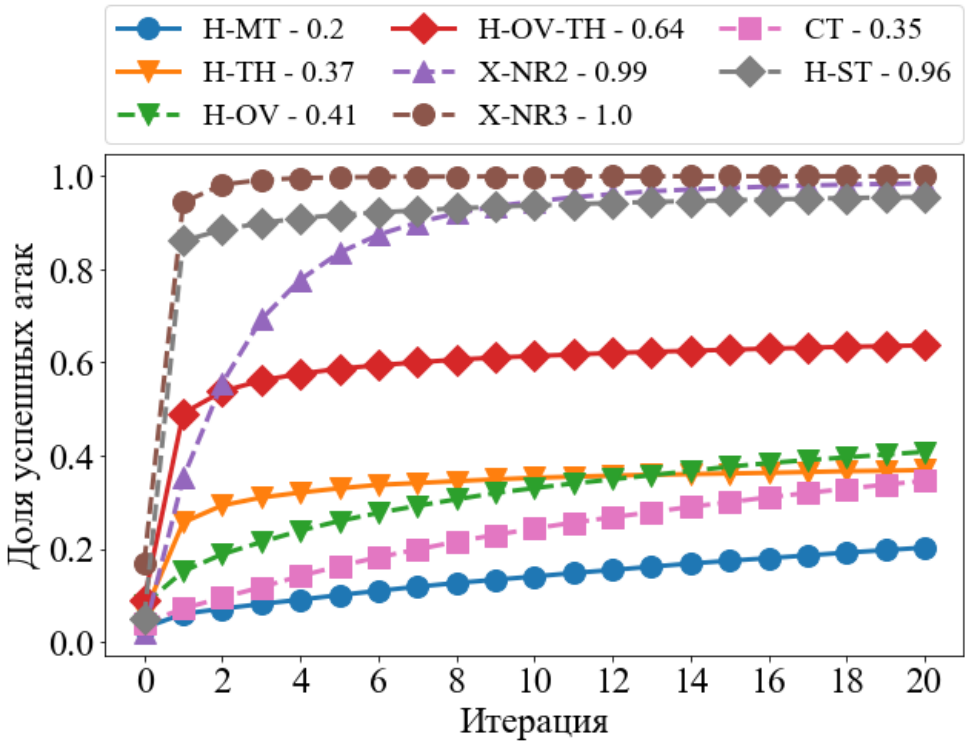


Рисунок 6.3 – График зависимости эффективности атаки от числа проделанных итераций алгоритма PGD. В качестве начального значения выбрана доля ошибок самой нейронной сети.

Это обосновывается тем, что в какой-то момент значения текущей модификации изображения окажутся на заданных границах и далее не могут быть изменены. Как видим наше предположение абсолютно верно для 5 из 8 задач классификации, для 3 из 8 кривая не достигла своего максимума. На этом графике можно заранее оценить количество итераций алгоритма PGD необходимых для проведения успешной атаки. Эта информация необходима для проведения устойчивого обучения нейронных сетей по методу PGD-k [8], время работы которого напрямую зависит от количества итераций применяемого алгоритма PGD. Для двух распространенных вариаций этого метода PGD-2 и PGD-7 можно сказать, что для первой в соответственно 3 из 8 и 5 из 8 задач повышение количества итераций существенно не повышает качество их работы.

6.6 Зависимость характеристик атаки от предсказанной вероятности оригинального изображения

Во время изучения результатов запуска экспериментов с алгоритмом PGD было обнаружено, что атакующие версии изображений с высокой предсказанной вероятностью принадлежности своему классу реже являются вредоносными. Это можно объяснить тем, что изображения, чья предсказанная вероятность достаточно велика, находятся в довольно большой области своего класса и найти успешную атакующую версию труднее. В результате была построена следующая зависимость:

1) Разобьем отрезок вероятностей [0.5, 1] на 10 равных частей.
2) Соотнесем каждое изображение в свою часть в соответствие с вероятностью принадлежности его классу. Т.е. изображение, результат предсказания которого равен [0.19, 0.81] отнесем к части [0.80, 0.85].
3) Посчитаем долю успешных атак алгоритмом PGD по отдельности в каждом под-отрезке.

Учитывая вышесказанное, ожидается, что доля успешных атак будет минимальна в последнем выделенном отрезке. Подтверждение этого предположения расширит наше представление о подпространстве собственного класса данных.

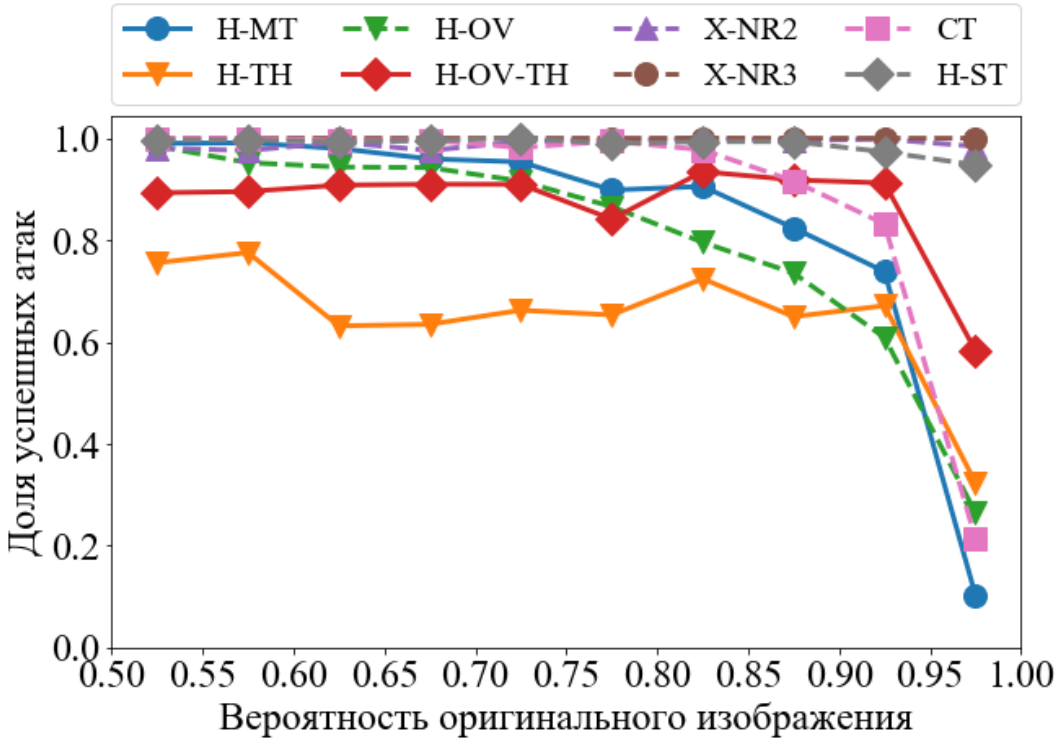


Рисунок 6.4 - График зависимости доли успешных атак от интервала исходной вероятности изображения

Для 6 из 8 задач классификации предполагаемая зависимость подтверждена, для 2 оставшихся эту зависимость сложно определить, поскольку практически все проводимые атаки были успешны и, следовательно, внутри каждого отрезка вероятностей это значение также почти 100%.

В отличие от алгоритма PGD алгоритмы CW и DF не ограничены в терминах нормы возмущения. Это позволяет рассматривать получаемые в результате их работы нормы возмущения как оценку размера множества текущего класса изображения. Построим зависимость аналогичную

вышеописанной, но вместо доли успешных атак внутри интервала посчитаем среднее значение L_2нормы возмущения (рисунок 6.5).

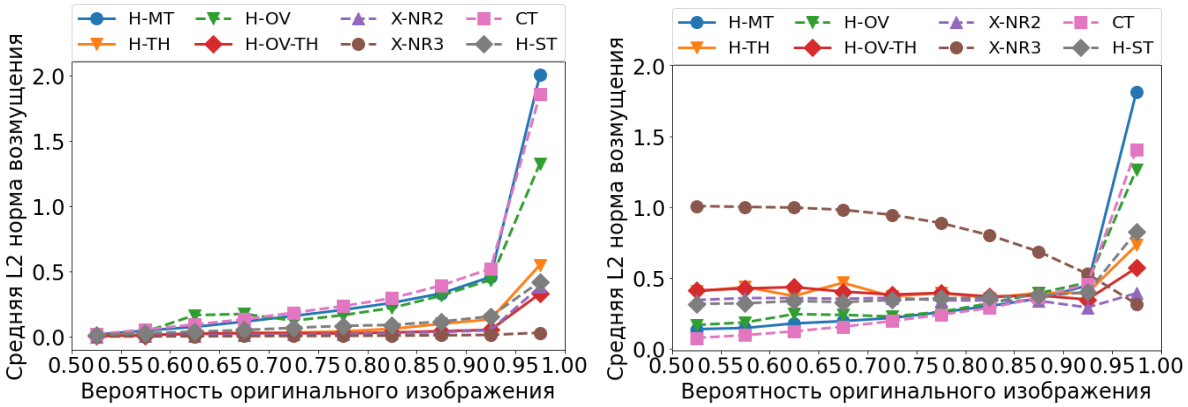


Рисунок 6.5 - График зависимости L_2нормы получаемого возмущения от интервала исходной вероятности изображения для алгоритмов Deepfool (слева) и CW (справа).

Как видим, для 7 из 8 наборов данных получаемые атакующие изображения находятся сильно дальше от исходных, в случае если его вероятность больше 0.95. Для набора данных X-NR3 алгоритм Deepfool находит невероятно маленькие возмущения во всех случаях, а поведение алгоритма CW по неопределенным причинам противоречит гипотезе.

ГЛАВА 7
ИССЛЕДОВАНИЕ АТАК ЧЕРНОГО ЯЩИКА

7.1 Методика проведения атак черного ящика

Как было сказано ранее атаки по методу черного ящика проводятся без использования информации об архитектуре сети или обученных весах. Нетрудно видеть, что в таких условиях генерация атакующих изображений, проведенная в главе 6, становится невозможной, поскольку все приведенные алгоритмы используют градиент функции выхода нейронной сети, который напрямую зависит от обученных параметров. В таком случае требуется принципиально новая методика проведения атак.

На сегодняшний день разработано несколько способов осуществления атак по методу черного ящика. Большинство из них основано на свойстве переносимости, упомянутом в главе 2: атакующее изображение, сгенерированное для атаки одной сети, часто успешно атакует другую сеть, обученную классифицировать изображения того же типа. Несмотря на то, что теоретических обоснований такому явлению практически нет, на практике оно неоднократно проявлялось. Полагаясь на это свойство можно сформулировать следующую методику действий:

1. На некоторой выборке изображений из предметной области классификации целевой сети обучаем свою “имитационную” сеть. В данной работе рассматривается режим, при котором доступна тренировочная выборка целевой сети - используем ее для тренировки своей сети.
2. Проводим атаку по методу белого ящика на обученную сеть - получаем атакующее изображение.

3. Подаем полученное изображение на вход целевой сети.

В результате проводим атаку по методу черного ящика, поскольку информация об архитектуре и весах целевой сети не использовалась, но только информация о соответствующих параметрах обученной имитационной сети.

Для проведения атак по такой методике в данной работе рассматриваются 5 архитектур глубоких нейронных сетей: InceptionV3, Densenet121, ResNet, Mobilenet, Xception. Из построенных в главе 5 задач классификации было выбрано 4, на основе которых проводилось исследование. Основываясь на методике выше для каждой выбранной задачи классификации выполнялась следующая последовательность действий:

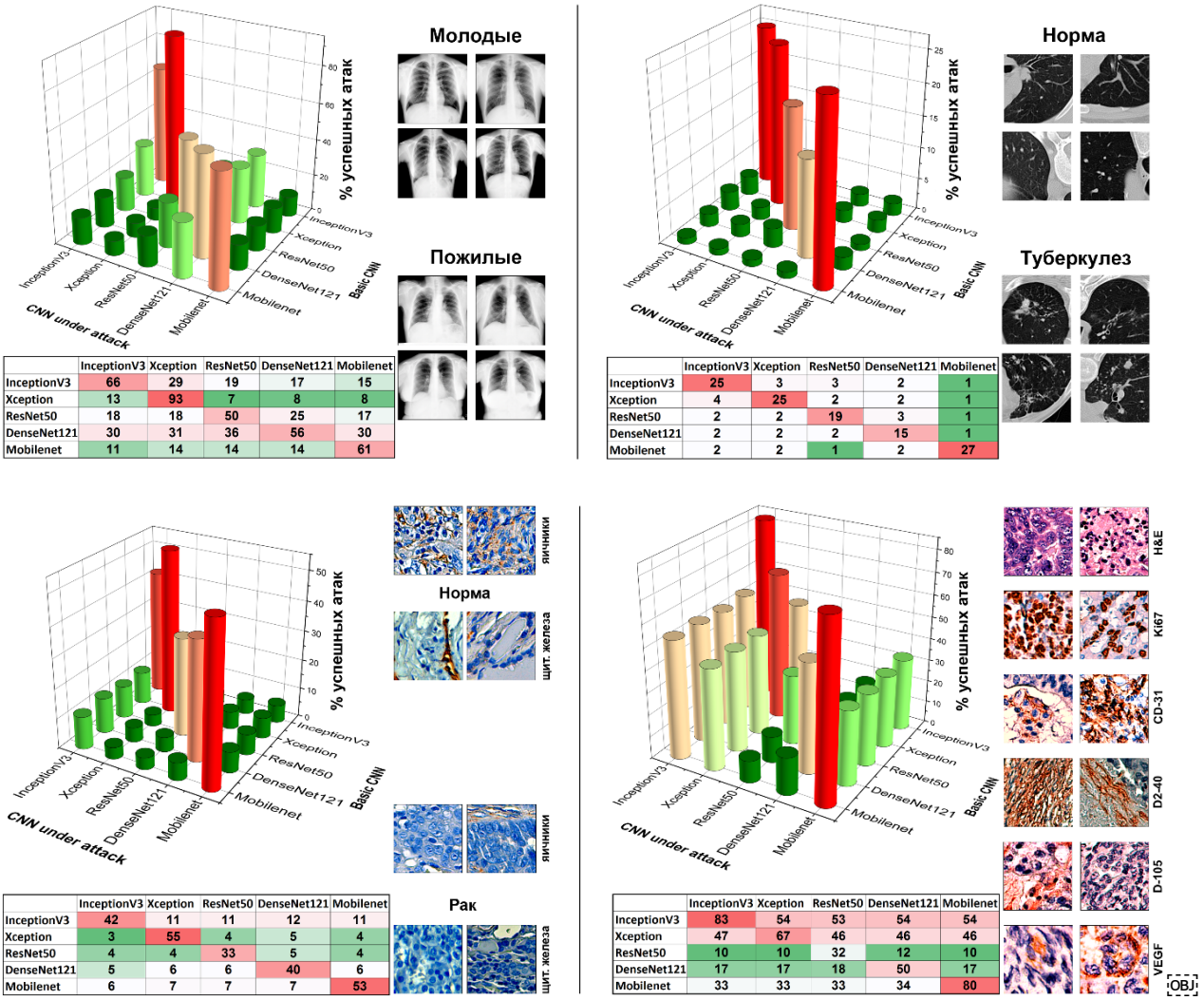
1. Обучаем каждую из приведенных архитектур сетей.
2. Каждую из обученных сетей атакуем по методу белого ящика, генерируя соответствующее атакующее изображение для каждого изображения из тестовой выборки, сгенерированные изображения сохраняем.
3. Для каждой пары обученных сетей одну сеть назначаем целевой, другую имитационной. Проводим атаку на целевую сеть путем подачи атакующих изображений, сгенерированных для имитационной, результаты предсказаний целевой сети сохраняем. Затем меняем целевую и имитационную сеть местами, проводим аналогичную атаку.

В качестве алгоритма генерации атакующих изображений был выбран PGD с ϵ равным 0.1.

7.2 Результаты проведения атак по методу черного ящика

В результате выполнения вышеописанной последовательности действий для каждой выбранной задачи классификации получили 25 наборов

вероятностей (результатов предсказаний): для каждой пары “целевая сеть-имитационная сеть” с различными сетями - 20 наборов, для каждой такой пары с одинаковыми сетями - 5 наборов (т.е. 5 наборов результатов атак по методу белого ящика, т.к. атаковалась та же сеть, для которой генерировались атакующие изображения). По полученным результатам вычислялась доля успешных атак.



	InceptionV3	Xception	ResNet50	DenseNet121	Mobilenet
InceptionV3	66	29	19	17	15
Xception	13	93	7	8	8
ResNet50	18	18	50	25	17
DenseNet121	30	31	36	56	30
Mobilenet	11	14	14	14	61

	InceptionV3	Xception	ResNet50	DenseNet121	Mobilenet
InceptionV3	25	3	3	2	1
Xception	4	25	2	2	1
ResNet50	2	2	19	3	1
DenseNet121	2	2	2	15	1
Mobilenet	2	2	1	2	27

	InceptionV3	Xception	ResNet50	DenseNet121	Mobilenet
InceptionV3	42	11	11	12	11
Xception	3	55	4	5	4
ResNet50	4	4	33	5	4
DenseNet121	5	6	6	40	6
Mobilenet	6	7	7	7	53

	InceptionV3	Xception	ResNet50	DenseNet121	Mobilenet
InceptionV3	83	54	53	54	54
Xception	47	67	46	46	46
ResNet50	10	10	32	12	10
DenseNet121	17	17	18	50	17
Mobilenet	33	33	33	34	80

Рис. 7.1 - Результаты проведения атак черного ящика для 4 задач классификации: трехклассовый набор рентгенов (X-NR3), КТ легкого (CT), четыре класса гистологии яичников и щитовидной железы (H-OV-TH), гистология с 6 химическими агентами (H-ST) . Показаны доли успешных атак в виде диаграммы и в виде таблицы. Также справа приведены примеры изображений из соответствующих наборов данных.

На рисунке 7.1 графически проиллюстрированы результаты проведения экспериментов. Можно заметить, что для двух задач классификации (CT, H-OV-TH) доля успешных атак черного ящика незначительна, для задачи X-NR3 заметно влияние таких атак, а для задачи H-ST сила атак черного ящика сравнима с атаками белого ящика (но, разумеется, меньше). Также стоит отметить, что доля успешных атак черного ящика зависит от того на вход какой сети подают сгенерированные изображения, но при этом зависимость от того для какой сети изображения генерировались не наблюдается.

ЗАКЛЮЧЕНИЕ

В данной работе было проведено широкое экспериментальное исследование состязательных атак в двух режимах доступности информации: в режиме белого ящика и в режиме черного ящика. По результатам исследования можно сделать следующие выводы:

- Проблема состязательных атак актуальна для задач анализа биомедицинских изображений: выбранные алгоритмы успешно атакуют обученные нейронные сети так, что их точность падает ниже 15%.
- Алгоритм PGD менее эффективен при тех же размерах возмущений изображения нежели алгоритмы DeepFool и CW, при этом с точки зрения L_2 нормы алгоритмы DeepFool и CW генерируют атакующие изображения близкого качества.
- Очень высокая вероятность принадлежности к классу (>0.95), предсказанная нейронной сетью, является хорошим показателем того, что сгенерированная для этого изображения атакующая версия либо будет неуспешна, либо будет сильнее отличаться от исходного.
- Для 3 из 4 выбранных задач классификации атаки по методу черного ящика с использованием алгоритма PGD показали незначительную эффективность.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Recht B., Roelofs R., Schmidt L., Shankar V.: Do CIFAR-10 Classifiers Generalize to CIFAR-10? arXiv preprint arXiv:1806.00451 (2018).
2. Akhtar N., Mian A.S.: Threat of Adversarial Attacks on Deep Learning in Computer Vision. IEEE Access 6, 14410–14430 (2018).
3. Litjens G., Kooi T., Bejnordi B., Setio A., Ciompi F., Ghafoorian M.: A survey on deep learning in medical image analysis. Medical Image Analysis 42, 60-88 (2017).
4. Ker J., Wang L., Rao J., Lim T.: Deep Learning Applications in Medical Image Analysis. IEEE Access 6, 9375-9389 (2018).
5. Zuxuan Wu, Ser-Nam Lim, Larry Davis, Tom Goldstein: Making an Invisibility Cloak: Real World Adversarial Attacks on Object Detectors. arXiv preprint arXiv:1910.14667 (2019).
6. Szegedy C., Wojciech Z., Sutskever I., Bruna J., Dumitru E., Goodfellow I., Fergus R.: Intriguing properties of neural networks. International Conference on Learning Representations (ICLR) 2014, pp. 1-10. Springer, Banff (2014).
7. Goodfellow I., Shlens J., Szegedy C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572v3 (2015).
8. Madry A., Makelov A., Schmidt L., Tsipras D., Vladu A.: Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv preprint arXiv:1706.06083v3 (2017).
9. Ozdag M.: Adversarial Attacks and Defenses Against Deep Neural Networks: A Survey. Procedia Computer Science 140, 152–161 (2018).
10. Xu W., Evans D., Qi Y.: Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. arXiv preprint arXiv:1704.01155v2 (2017).
11. Wang H., Yu Chun-Nam: A Direct Approach to Robust Deep Learning Using Adversarial Networks. arXiv preprint arXiv:1905.09591v1 (2019).
12. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In Security and Privacy (SP). IEEE Symposium 39–57, (2017).
13. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013).

14. Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard: DeepFool: a simple and accurate method to fool deep neural networks. arXiv preprint arXiv:1511.04599v3 (2015).