**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
|  | **ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ**  **Кафедра**  **«Криптология и кибербезопасность»** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ОТЧЕТ

О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

ИССЛЕДОВАНИЕ СПОСОБОВ СОВЕРШЕНИЯ ADVERSARIAL-АТАК НА НЕЙРОННЫЕ СЕТИ.

|  |  |  |
| --- | --- | --- |
| Исполнитель:  студент гр. Б16-505 | \_\_\_\_\_\_\_\_\_\_\_ (подпись, дата) | Свинин В.П. |
| Научный руководитель: | \_\_\_\_\_\_\_\_\_\_\_ (подпись, дата) | Бурлаков М.Е. |
| Зам. зав. каф. № 42: | \_\_\_\_\_\_\_\_\_\_\_ (подпись, дата) | Когос К.Г. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва – 2019**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

РЕФЕРАТ

СОДЕРЖАНИЕ

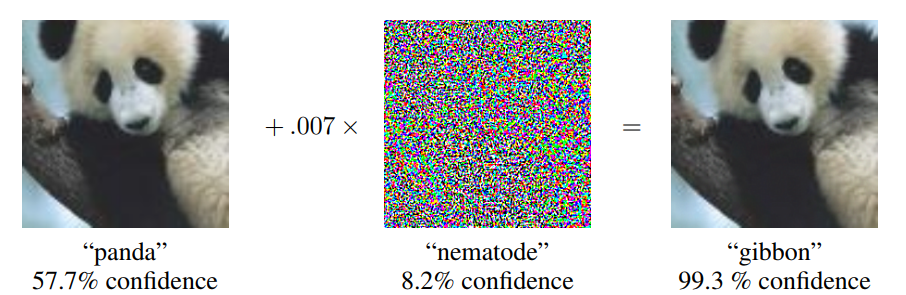
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете применяются следующие термины с соответствующими определениями, обозначениями и сокращениями, используемые для описания adversarial-атак на системы глубокого обучения, решающие задачи компьютерного зрения. В следствие отсутствия устоявшейся терминологии, некоторые термины не были переведены и используются в изначальном виде. Пояснение к ним дано в скобках.

* Adversarial-атака – от англ. «adversary» – противник, соперник.
* Adversarial-пример/изображение – намеренно искаженная (например, путём добавления шума) версия исходного изображения, используемая для обмана алгоритма машинного обучения.
* Adversarial-искажение – шум, добавляемый к исходному изображению для того, чтобы сделать из него adversarial-пример.
* Adversarial-обучение – добавление adversarial-примеров к обучающей выборке с целью повышения устойчивости модели adversarial-атакам.
* Black-box атака – атака на выбранную модель, совершаемая без знания её структуры. Иногда допускается, что атакующий обладает ограниченными знаниями о структуре модели (например, о её процессе обучения или архитектуре), но не знает точные значения её параметров. В этом случае используется термин *gray-box* или *semi-black-box* атака.
* Детектор – позволяет определить, является ли изображение adversarial-примером.
* Fooling ratio (доля успешных обманов) – доля изображений, на которых модель меняет результат предсказания после добавления к ним adversarial-искажений.
* Одношаговые методы – методы создания adversarial-искажений, предполагающие однократное вычисление, например, вычисление градиента функции потерь модели. Напротив – итеративные методы, предполагающие совершение множества последовательных однотипных операций и являющиеся, как правило, вычислительно более сложными.
* Квазинезаметные искажения – искажения, практически не изменяющие изображение для человека.
* Targeted-атаки (направленные атаки) – используются для того, чтобы модель неверно выбрала целевой класс (от англ. «target» - цель). Их противоположность – non-targeted-атаки (ненаправленные атаки), цель которых – заставать модель сделать любое неверное предсказание.
* Модель угроз – описывает типы потенциальных атак, в соответствии с моделью их совершения (например, black-box/white-box атака, направленная/ненаправленная атака).
* Transferability (переносимость) – отражает возможность использования adversarial-примеров на моделях, которые не были использованы при его создании.
* Универсальное искажение (UAP – universal adversarial perturbation) – искажение, имеющее высокую вероятность обмануть модель на любом выбранном изображении. Следует отметить, что универсальность – это свойство независимости искажения от класса изображения, в то время как переносимость – независимость искажения от используемой модели.
* White-box-атака предполагает, что у атакующего имеются полные сведения о модели, включая её параметры, архитектуру, метод обучения и обучающую выборку.

ВВЕДЕНИЕ

Глубокое обучение заметно продвинуло машинное обучение в задачах компьютерного зрения (таких как 157классификация, распознавание и сегментация), задачах распознавания и генерации речи, машинного перевода и др. Последние годы заметно возросла доля систем, полагающихся на глубокое обучение, например, беспилотные автомобили, использующие системы глубокого обучения для навигации на дороге [49], [50]; дроны и роботы [51], [52]; банки использующие системы распознавания лиц в банкоматах [53]; системы распознавания вирусов, основанные на поиске аномалий в поведении с помощью глубокого обучения [1]; системы распознавания голоса [54]. Очевидно, что системы, полагающиеся на глубокое обучение, особенно в задачах компьютерного зрения, постепенно занимают всё более важную роль в повседневной жизни. Множество примеров использования глубокого обучения поднимает важные вопросы, связанные с безопасностью и защищенностью этих систем. Недавние исследования показывают, что системы глубокого обучения могут быть уязвимыми к специально сгенерированным входным данным, называемыми adversarial-примерами, остающимся для человека внешне неотличимым от исходных. Использование таких примеров может заставить классификатор (нейронную сеть) поменять своё мнение об исходном примере, при этом внушая высокую уверенность в неправильном решении. Более того, один и тот же adversarial-пример способен обмануть множество не связанных между собой cетей. Пример такого adversarial-изображения, нацеленного на обман классификатора, приведён на рисунке 1 – после добавления к исходной картинке adversarial-искажения, предсказание сети меняется с «панда» (с уверенностью 57.7%) на «гиббон» (99.3%)

  
Рисунок 1 — Adversarial-изображение

В данном отчете приведено описание и классификация основных подходов создания adversarial-примеров и заданы основные условия и ограничения, определяющие возможность совершения с их помощью adversarial-атак.

**2 Классификация adversarial-атак**

Классификацию подходов, применяемых для создания adversarial-примеров, можно провести исследуя их различия в трёх направлениях: модель угроз, adversarial-искажение и атакуемая модель/набор данных.

2.1 Модель угроз

В соответствии с работой Papernot и др. [1], все рассматриваемые adversarial-атаки объединены следующим набором свойств:

* Модель может быть атакована только на этапе тестирования, при этом атакующих способен изменять входные данные только после завершения обучения модели. Атакующий может иметь некоторую информацию о модели (такую как архитектура или параметры), но не способен изменять саму модель. Атаки на процесс обучения представляют собой отдельную тему для исследований и не будут рассмотрены в данной работе [62], [63].
* В работе рассматриваются атаки на модели, построенные на основе нейронных сетей.
* Целью атакующего является нарушение целостности модели, выраженное в виде падения значения целевой метрики (доли успехов, F-меры и др.).

Далее, в зависимости от условий совершения, adversarial-атаки могут быть охарактеризованы типом вызываемой ошибки, наличием у атакующего информации о модели, направленностью атаки и количеством вычислительных шагов для создания adversarial-примера.

2.1.1 Тип вызываемой ошибки

* Атаки, вызывающие ошибки первого рода (атакующий создаёт adversarial-пример, вызывающий ложной срабатывание – false positive). Например, в задаче распознавания вредоносных программ, ложным срабатыванием будет принятие обычной программы за зловредную. В задаче классификации за ложное срабатывание принимается случай, когда искаженное изображение неправильно классифицируется человеком (или не несёт для него смысла), при этом не меняя предсказание модели. Пример ошибки первого рода проиллюстрирован на рисунке 2 – нейронная сеть, обученная на ImageNet, показывает высокую уверенность (более 99.6%) при классификации изображений [2].
* Атаки, вызывающие ошибки второго рода (атакующий создаёт adversarial-пример, вызывающий пропуск события – false negative). При распознавании вредоносных программ ошибки этого вида приведут к пропуску вредоносной программы, а при классификации, adversarial-пример, легко классифицируемый человеком, не будет распознан моделью (или вызовет высокую уверенность в ложном классе).

  
Рисунок 2 — Пример ошибки первого рода

2.1.2 Наличие информации о модели у атакующего

В зависимости от полноты информации об атакуемой модели, adversarial-атаки можно разделить на *white-box* и *black-box*.

* В случае white-box атаки, атакующему известна вся информация о модели: её параметры и архитектура, процесс обучения, гиперпараметры и использованная обучающая выборка. Большинство атак этого типа для генерации adversarial-примеров напрямую используют градиенты параметров модели.
* В случае block-box-атаки, атакующий выступает в роли простого пользователя, которому известна лишь реакция модели на входные данные (метка класса или его вероятность). Допустим вариант gray-box атаки, в которой у атакующего имеется ограниченная информация о модели (например, гиперпараметры, использовавшиеся в процессе обучения). Этот тип не выделяется в отдельный класс, т.к. дополнительной информации недостаточно для перехода к white-box атаке, и может быть классифицирован как black-box атака с дополнительными сведениями о модели.

2.1.3 Направленность атаки

* Targeted атака (направленная атака) стремится заставить модель выбрать целевой класс, даже если он является неправильным. Например, если нейронная сеть, обученная на MNIST, на изображении из тестового набора выдаёт результат «0» и требуется изменить его на «7», то применяется *targeted* атака.
* Non-targeted атака (ненаправленная атака) стремится добиться любого неправильного результата (в случае бинарной классификации, атаки этих типов совпадают). Например, при совершении non-targeted атаки на систему распознавания лиц, атакующему стремится быть идентифицированным как любой другой человек. Ненаправленные атаки осуществляются проще, чем направленные, т.к. у атакующего имеется больше свободы в поиске adversarial-примеров. Обычно, ненаправленные атаки осуществляются одним из двух способов:   
  1) путём создания нескольких adversarial-примеров (с помощью направленной атаки) и выбором одного из них (например, наиболее похожего на исходное изображение);   
  2) путём минимизации вероятности, приписанной классификатором корректному классу.

2.1.4 Зависимость от количества шагов вычислений

* Одношаговые (one-step) атаки предполагают однократное вычисление при поиске adversarial-примера, например, вычисление градиента функции потерь модели (FGSM).
* Итеративные методы атак предполагают многократное последовательное изменение исходного примера до превращения его в adversarial-пример. Часто атаки этого типа являются вычислительно более сложными, но приводят к лучшим результатам. Для некоторых задач (например, для обучения с подкреплением) атаки этого типа могут быть неприменимы в связи с трудностью вычисления adversarial-искажений.

2.2 Adversarial-искажение

Поиск малых искажений, незаметных для человека, но заметно влияющих на предсказание модели является основной задачей adversarial-атак. Adversarial-искажения можно классифицировать, ориентируясь в трёх направлениях: универсальность искажения, способ его ограничения и используемая при этом норма.

* В зависимости от универсальности созданного искажения, атаки можно разделить на: 1) *индивидуальные*, создающие различные искажения для различных исходных примеров; 2) *универсальные*, создающие искажение, которое может быть успешно применено ко всем исходным примерам вне зависимости от их класса. Следует заметить, что в условиях реальных систем использование универсальных adversarial-искажений заметно упрощает задачу атакующего.
* Adversarial-искажения могут являться как результатом, так и ограничением оптимизационной проблемы. При оптимизации самого искажения атакующий стремиться уменьшить его величину так, чтобы оно было незаметно для человека, в то время как использование искажения в качестве ограничения просто требует от искажения быть достаточно малым.
* Оценку величины искажения можно производить, основываясь на его -норме:

Наиболее распространенными нормами являются , и :

– -норма равняется количеству пикселей, измененных в исходном изображении при создании adversarial-примера.

– -норма – евклидово расстояние между исходным и adversarial-изображением.

– -норма равняется модулю максимального значения изменения пикселей исходного изображения.

2.3 Атакуемая модель/набор данных

Исследователи оценивают эффективность adversarial-атак основываясь на разных атакуемых моделях и наборах данных, используемых для их обучения, что может привести к появлению неоднозначности. Поэтому при оценке эффективности необходимо учитывать:

* Использованный набор данных. MNIST [25], CIFAR-10 [20] и ImageNet [56] являются наиболее популярными наборами данных для оценки эффективности adversarial-атак, хотя в последнее время, из-за недостаточного объема и сложности, MNIST и CIFAR-10 становятся всё менее популярными. Для правильной оценки adversarial-атак требуется хороший набор данных.
* Атакуемая модель. Наиболее популярными архитектурами моделей для проведения атак являются LeNet [57], VGG [58], AlexNet [59], GoogLeNet [40], CaffeNet [61] и ResNet [62].

**3 Известные adversarial-атаки**

В этом разделе рассмотрены наиболее известные методы атак на нейронные сети в «лабораторных условиях»; их эффективность оценивается на стандартных наборах данных (MNIST, CIFAR-10 или ImageNet) в типичной задаче компьютерного зрения – классификации. Следует заметить, что подходы, рассмотренные в данном разделе, составляют основу для атак в реальных условиях и каждый из них имеет потенциал повлиять на развитие безопасности глубокого обучения в практических задачах.  
 Обзор в данном разделе ведётся, в основном, в хронологическом порядке, к основным подходам и главным направлениям атак даётся краткое техническое описание, в то время как остальные подходы описаны в общих чертах, подробности о них читатель может найти в приложенных отсылках к исходным работам.

3.1 Box-constrained L-BFGS (L-BFGS)

Впервые существование adversarial-искажений продемонстрировал Szegedy и др. в [3].   
Пусть исходное изображение представлено в виде вектора , индекс «c» подчеркивает, что изображение «чистое» («clear»). Чтобы найти искажение , прибавление которого к исходному изображению изменит результат классификации, визуально не изменяя изображение, было предложено найти решение следующей задачи:

Авторы предложили найти нетривиальное решение (1), в котором предсказание классификатора отличается от исходного класса изображения. В этом случае нахождение становится нетривиальной проблемой, и для нахождения приблизительного решения авторы предлагают использовать алгоритм L-BFGS: находится минимальное для которого выполняется условие:

В случаях, когда функция потерь классификатора представляет собой выпуклую функцию, решение (2) даёт единственное решение. Но, в случае глубоких нейронных сетей, это, чаще всего, не так.  
 Для получения adversarial-примера, вычисленное искажение просто прибавляется к исходному изображению. Как показано на рисунке 2, полученные adversarial-изображения визуально неотличимы от исходных изображений. Также было замечено, что искажения, вычисленные для нейронной сети могут работать и для других сетей.

Во время публикации работы считалось, что нейронная сеть выделяет признаки, определяющие пространство, где воспринимаемые между изображениями различия приблизительно равняются евклидовому расстоянию между ними. Эти противоречивые результаты пробудили интерес исследователей к adversarial-атакам.

3.2 Fast Gradient Sign Method (FGSM)

Szegedy и др. в [3] было замечено, что устойчивость нейронных сетей к adversarial-примерам может быть повышена за счёт adversarial-обучения. Для того, чтобы повысить эффективность adversarial-обучения, Goodfellow и др. в [4] бы разработан метод быстрого расчёта adversarial-искажений:

градиент функции стоимости в точке, определяемой параметрами модели , по отношению к исходному изображению ,скаляр, ограничивающих размер искажения.

Adversarial-примеры, полученные с помощью FGSM, используют «линейность» нейронных сетей в многомерном пространстве, хотя в то время такие модели считались сильно нелинейными. Goodfellow и др. в [4] выдвинули гипотезу, что дизайн современных нейронных сетей, намеренно поощряющий линейное поведение для вычислительной простоты (ReLU), делает их уязвимыми к простому поиску подобных искажений. Эта идея получила название «гипотезы линейности» и поддерживается существованием FGSM похода.  
 Kurakin и др. [5] отмечают, что adversarial-примеры, полученные с помощью FGSM для изображений из ImageNet, достигают успешного обмана сетей (top-1 error rate) в 63-69% случаев при . Также авторы предложили «одношаговый направленный FGSM метод», в котором, вместо использования в (3) исходного класса , вычисляется градиент по отношению к классу , имеющему наименьшую вероятность среди предсказанных сетью для изображения

Полученное таким образом искажение вычитается из исходного изображения, чтобы сделать из него adversarial-изображение. Для нейронной сети, использующей кросс-энтропию как функцию потерь, такая операция максимизирует вероятность выбора в качестве класса adversarial-примера. Предполагается, что использование случайного класса в качестве также может привести к успешному обману сети, хотя это может привести к менее интересным результатам, например к классификации одной породы собак в качестве другой. Авторы также показали, что использование adversarial-обучения повышает устойчивость сети к атакам с помощью FGSM и его вариантов.

FGSM искажает изображение, увеличивая значение функции потерь для правильного класса. Функция позволяет максимизировать величину искажения, в то время как ограничивает его равномерную норму (-норму). Miyato и др. [7] предложил похожий метод вычисления adversarial-искажений:

В приведенном выражении, градиент нормализуется делением на -норму. Kurakin и др. [5] назвали этот способ «FGL2» методом и предложили использовать для нормализации равномерную норму, назвав получившуюся технику «FGL∞». Все эти методы рассматриваются как «одношаговые» методы вычисления adversarial-искажений.

3.3 Basic & Least-Likely-Class Iterative Methods (BIM & ILCM)

Одношаговые методы создают adversarial-изображения делая один большой шаг в направлении увеличения значения функции потерь классификатора. Интуитивное продолжение этой идеи – итеративное совершение множества маленьких шагов, сопровождающееся корректировкой направления движения на каждом шаге.  
 Простой итеративный метод (BIM) [6] предполагает именно это:

где – искаженное изображение на -ой итерации, - ограничивает значения пикселей изображения на , – определяет размер шага (обычно, ).

Алгоритм начинается с и совершает количество итераций, определяемое выражением . Было замечено [8], что BIM является эквивалентом PGM (Projected Gradient Descent) – стандартному методу оптимизации выпуклых функций.

По аналогии с модификацией FGSM до его «направленной» версии, Kurakin и др. [6] предложили расширить BIM до Iterative Least-likely Class Method (ILCM). В этом случае, в качестве класса в (5) предлагается использовать класс (с наименьшей вероятностью, данной классификатором). Было показано, что adversarial-примеры, полученные с помощью ILCM, сильно влияют на точность классификации сетей с архитектурой Inception v3, даже для малых значений .

3.4 Jacobian-based Saliency Map Attack (JSMA)

Наиболее частый подход к генерации квазинезаметных adversarial-изменений – это ограниение их или -норм. Papernot и др. [9] предложили способ проведения атак, путём ограничения -норм adversarial-искажения. На практике, это означает что такие искажения модифицируют только несколько пикселей исходного изображения. Основой их алгоритма является поочередная модификация пикселей исходного изображения и отслеживание изменений в результирующей классификации. Изменения отслеживаются путём построения карты градиентов выходных слоёв сети – в ней для каждого пикселя исходного изображения задаётся значение – большее значение соответствуют большей вероятности изменения предсказанного класса при изменении данного пикселя. После этого, алгоритм выбирает пиксель с наибольшей вероятностью обмануть сеть и изменяет его. Этот процесс повторяется до тех пор, пока не достигается максимальное количество измененных пикселей или пока не удастся обмануть классификатор.

3.5 One Pixel Attack

Крайним случаем adversarial-атаки можно считать изменение единственного пикселя исходного изображения. Su и др. [10] провели атаку на три разные сети с успехом на 70.97% тестовых изображений, изменяя значения всего лишь одного пикселя. Было также показано, что средняя уверенность классификатора на ложных примерах равнялась 97.47%.

Adversarial-примеры были получены с использованием метода дифференциальной эволюции [11]: были выбраны 400 векторов, содержащих xy-координаты и RGB значения случайных пикселей исходного изображения. Затем, значения векторов были случайным образом изменены, создавая следующее поколение, при этом потомок соревнуется со своим родителем, используя в качестве целевого критерия вероятность, полученную от классификатора. После нескольких итераций, для изменения изображения выбирается последний оставшийся потомок.

Даже с такой простой эволюционной стратегией авторам [10] удалось провести успешные атаки на нейронные сети. Следует заметить, что дифференциальная эволюция позволяет генерировать adversarial-примеры без доступа к параметрам и градиентам сети. Единственная необходимая информация – это вероятности, полученные от классификатора.

3.6 Carlini and Wagner Attacks (C&W)

При появлении методов защиты простив adversarial-изменений, Carlini и Wagner [42] предложили набор из 3 атак, делающих изменений квазинезаметными путём ограничения их и -норм. Ими было показано, что данные атаки способны преодолеть механизм защиты, называемый defensive distillation [12][13]. Более того, было показано, что adversarial-примеры, полученные для незащищенных этим механизмом сетей, могут быть перенесены на защищенные, что допускает совершение black-box-атак.

В то время, когда наиболее частым методом совершения black-box-атак является использование переносимости adversarial-примеров, Chen и др. [14] предложили «Zeroth Order Optimization» атаку, заключающуюся в приблизительном вычислении градиентов атакуемой модели. Для обхода прямого вычисления градиентов (что требует больших вычислительных затрат), авторы предложили использовать алгоритм ZOO-ADAM, основанный на оптимизаторе ADAM [73], позволяющий итеративно обновлять выбранное случайным образом искажение. Авторы показали, что ZOO достигает результатов, сравнимых по успешности с методом C&W.

3.7 DeepFool

Moosavi-Dezfooli и др. [15] предложили вычислять квазинезаметные искажения итеративным способом. Их алгоритм заключается в линеаризации функции потерь и вычисления минимального искажения, необходимого для изменения классификации при данной линеаризации. На каждом шаге алгоритма функция потерь вычисляется при добавлении полученного на прошлом шаге искажения. Так происходит до тех пор, пока классификатор не станет выдавать ошибочный результат (таким образом DeepFool можно отнести к ненаправленным атакам). Авторы [15] показывают, что полученные adversarial- искажения имеют меньшие нормы, чем полученные с помощью FGSM и имеющие тот же процент успешных обманов.

3.8 Universal Adversarial Perturbations (UAP)

В то время, как FGSM, ILCM, DeepFool и др. методы нацелены на вычисление adversarial-изменений, обманывающих классификатор на единственном изображении, «универсальные» adversarial-искажения, вычисленные Moosavi-Dezfooli и др. [16], с высокой вероятностью подходят для любого изображения. Важно, что эти искажения могут быть квазинезаметными для человека. Для их формального определения предположим, что исходные изображения получены из распределения Ω. Тогда изменение можно назвать «универсальным», если оно удовлетворяет следующему ограничению:

где – вероятность, – процент успешных обманов, -норма,

– константа, определяющая величину искажения.

Строго говоря, искажения, удовлетворяющие (6) должны называться   
«-универсальным», подчёркивая сильную зависимость от названных параметров. Как бы то ни было, в литературе их обычно называют просто «универсальными».

Авторы [16] вычислили универсальные adversarial-искажения, ограничивая их и -норму, и показали, что искажения в 4% от нормы исходного изображения достигают значительного процента успешных обманов (80%) на лучших классификаторах. Их метод похож на метод, используемый в DeepFool – последовательный сдвиг изображения к границе класса, определенной классификатором. Но, в этом случае, все изображения из тренировочной выборки последовательно сдвигаются к своим границам, накапливая суммарное искажение, ограничиваемое по -норме величиной .

Алгоритм, предложенный в [16], вычисляет adversarial-искажения, нацеленные на единственную модель. Тем не менее, было показано, что эти искажения переносятся и на другие сети, особенно на имеющие сходную архитектуру. В этом отношении, авторы [16] называют искажения «вдвойне универсальными». Более того, ими было показано, что высокого процента успешных обманов (более 50%) можно достичь на тренировочной выборке, состоящей всего лишь из 2000 изображений.

Другие методы получения универсальных искажений были предложены   
Khrulkov и др. [17] (позволяет получать хорошие результаты, используя маленькое количество изображений), Mopuri и др. [18] (получили универсальные искажения, независимые от данных).

3.9 UPSET и ANGRI

Две black-box-атаки были предложены Sarkar и др. [19].   
UPSET (Universal Perturbations for Steering to Exact Targets): для n классов алгоритм вычисляет n независимых от изображения искажений, соответствующих направленному изменению классификации для любого изображения. UPSET основан на генерирующей сети , принимающей на вход выбранный класс и создающей направленное adversarial-искажение . Полностью, UPSET-сеть решает следующую оптимизационную проблему:

где значения пикселей изображения нормализованы в промежутке , а s – скаляр.

ANGRI похожим образом вычисляет индивидуальные для каждого изображения искажения. Авторами было показано, что оба алгоритма достигают высоких показателей на выборках из MNIST и CIFAR-10.

3.10 Houdini

Cisse и др. [21] предложили алгоритм для обмана основанных на градиентном обучении подходов, подходящий для различных функций потерь. Типичные adversarial-атаки предполагают наличие дифференцируемой функции потерь, что иногда может не соблюдаться. Например, функция потерь в задаче распознавании речи может быть основана на доле правильно разобранных слов, что не предполагает прямого использования градиента. Houdini используется для создания adversarial-примеров для таких задач. Кроме удачного создания adversarial-примеров для классификации, Houdini был использован для совершения атаки на систему распознавания речи [22]. Также авторами была продемонстрирована переносимость атак на распознавание речи, путём совершения black-box-атаки на Google Voice.

3.11 Adversarial Transformation Networks (ATNs)

Baluja и Fisher [23] обучили нейронные сети создавать adversarial-примеры для других нейронных сетей. Adversarial-примеры, созданные этими сетями, вычисляются путём минимизации составной функции потерь: первая часть обеспечивает квазинезаметность adversarial-искажений, а вторая направлена на изменение классификации результирующего изображения.

Hayex и Danezis [24] также использовали нейронную сеть для создания adversarial-примеров для black-box атак. В представленных результатах, adversarial-примеры, созданные сетью остаются визуально неотличимыми от исходных изображений, но уменьшают точность классификации с 99.4% до 0.77% на MNIST и с 91.4% до 6.8% на CIFAR-10.

3.12 Другие атаки

Sabour и др. [26] описали вероятность создания adversarial-примеров путём изменения начальных слоёв нейронной сети. Авторы показали, что возможно изменить внутреннее представление adversarial-изображения, повторяющее представление изображений различных классов. Papernot и др. в [27] изучили переносимость adversarial-атак. Narodytska и Kasiviswanathan в [28] представили способ совершения black-box-атаки, путём изменения нескольких пикселей изображения. Liu и др. в [29] представили «эпсилон-окрестность»-white-box-атаку, успешность которой по отношению к защищенным через defensive distillation сети достигла 100%. Oh и др. в [30] применили теорию игр для разработки стратегии противодействия существующим способам защиты от adversarial-атак. Mpouri и др. в [18] разработали метод вычисления универсальных искажений. Hosseini и др. в [31] ввели понятие «семантические-adversarial-примеры» - изображения, которые обозначают одинаковые объекты для человека, но разные для классификатора. Kanbak и др. [32] предложили ManiFool – алгоритм, измеряющий устойчивость нейронной сети к геометрически искаженным изображениям. Dong и др. [33] предложил итеративный метод совершения adversarial-black-box-атак. Недавно, Carlini и Wagner [34] показали, что десять методов защиты от adversarial-атак могут быть преодолены, используя новые атаки, использующие измененные функции потерь. Rozsa и др. [35] предложили метод создания нескольких различных adversarial-примеров для исходного изображения. Следует упомянуть, что adversarial-искажения можно использовать для улучшения качества классификации. Yoo и др. в [36] предложили улучшение классификации путём добавления небольших искажений к исходному изображению.

**4 Adversarial-атаки в реальных условиях**

В отличие от атак в «лабораторных условиях», описанных в разделе 3.1, модель угроз атак в реальном мире налагает множество ограничений на возможности атакующего. Основным из них является факт того, что атакующий не имеет прямого доступа к входам модели и вынужден обращаться к ней через определенные интерфейсы (например, через видеокамеру). Таким образом, он не может нарушить целостность работы модели напрямую, но имеет возможность изменять воспринимаемый камерой объект. Цель атакующего остаётся той же – изменить предсказание модели.

Как и атаки в «лабораторных условиях», атаки в реальном мире могут проводиться в black-box и white-box режиме. Распространённым методом атаки является обучение модели-близнеца основываясь на доступных предсказаниях black-box модели и последующая генерация adversarial-примеров с её помощью (путём проведения против неё white-box атаки). Благодаря свойству переносимости [27], созданные таким образом adversarial-примеры должны сработать и на исходной модели.

4.1 Детекторы adversarial-примеров

Интерфейсы атакуемой системы могут накладывать ещё больше ограничений на атакующего, например, разработчики модели могут установить детекторы adversarial-изображений перед входом в модель для предотвращения потенциальных атак. Примером такого детектора может служить оценка объёма в системе распознавания лиц, позволяющая отличить реального человека от напечатанной фотографии. Для обхода подобных механизмов, атакующий, обычно, стремится найти объект, который проходит эту проверку, а затем накладывает на него adversarial-искажения. Тем не менее, существуют подходы, позволяющие определить их наличие. Xu и др. [65] предложили так называемый “feature squeezing”, заключающийся в уменьшении глубины цвета изображения и его сглаживания. Измененное таким образом изображение, вместе с исходным, передаётся в модель, затем полученные предсказания сравниваются. При большом различии между ними изображение считается adversarial-примером. Было показано [65][66], что такой метод является эффективным как против классических подходов (таких как FGSM [4]), так и против более сильных атак (C&W [42]). Meng и Chen [68] предложили добавить дополнительную сеть-детектор (названную MagNet), задачей которой является классификация изображений как «чистых» или adversarial-примеров. В ходе обучения, сеть-детектор стремиться выучить форму пространства «чистых» примеров, чтобы в ходе реальной работы вычислять adversarial-примеры, расположенные за его границей. Подобная защита показала себя эффективной против FGSM, BIM/ILCM, DeepFool и C&W, при этом восстанавливая точность классификации с 0% до исходных значений для двух последних атак.

Другим распространённым подходом к детекции adversarial-примеров является оценка активации элементов сети. Lu и др. [64] предложили добавить SVM-классификатор (названный ими SafetyNet) к значениям ReLU элементов в последних слоях сети, предполагая, что adversarial-изображения вызывают в них необычные паттерны активации. Такой подход оказался эффективным при обнаружении adversarial-примеров, полученных с помощью FGSM [4], DeepFool [15] и итеративных атак [6]. Похожая система была предложеа Li и др. [67], которые оценивали активации фильтров конволюционной сети. Их подход позволил обнаружить более 85% adversarial-примеров, созданных с помощью L-BFGS атак.

4.2 Ограничения, типичные для атак в реальных условиях  
Обычно, проведение adversarial-атаки состоит из двух шагов: 1) описание функции потерь для оптимизатора adversarial-искажения; 2) поиск adversarial-искажения и его применение. В данном разделе представлены ограничения, которые вызваны точностью adversarial-искажений и могут встретиться при совершении атак на практике.

4.2.1 Точность нанесения adversarial-искажений

Одной из проблем, встающих перед атакующим, является нанесение adversarial-искажений на атакуемый объект. В идеальных условиях пиксели искажения накладываются на пиксели исходного изображения, полностью перекрывая их. В реальных условиях покрытие атакуемого объекта произвольным цветом сильно затруднено. Это влечёт за собой ограничение, накладываемое на метод получения adversarial-искажения – он должен находить искажение, которое может быть без потерь нанесено на атакуемый объект. Это ограничение тесно связано со способом, которым атакующих собирается вносить данное изменение. Например, если он планирует печатать adversarial-искажение на бумаге и наклеивать её на атакуемы объект, то цвета adversarial-искажения должны принадлежат цветовому профилю используемого принтера. Если же атакующий планирует проецировать adversarial-искажения, используя монохроматический источник света, то искажения должны быть ограничены определенным цветом.

Для обеспечения выполнения этих условий, атакующий может обновить функцию потерь оптимизатора искажения, добавив слагаемое, отражающее сложность его нанесения на атакуемый объект. Такой подход, названный NPS (non-printability score), впервые был предложен Sharif и др. в [69]:

В соответствии с определением (8), значение NPS будет мало, когда цвет пикселя близок к множеству доступных цветов . Авторами [41], при создании adversarial-искажений для дорожных знаков, была также учтено то, что нанесённое искажение должно быть упрощено, то есть в нём должно быть использовано как можно меньше уникальных цветов:

– множество уникальных цветов (определенных в виде RGB-троек).  
С добавлением этого слагаемого к функции потерь, оптимизатор будет искать удобное для печати adversarial-искажение.

4.2.2 Точность adversarial-искажения

В «лабораторных условиях» атакующих имеет возможность независимо изменять значения отдельных пикселей adversarial-искажения. В условиях реальных атак изменения воспринимаются через видеокамеры атакуемой системы и у атакующего нет возможности напрямую контролировать значения пикселей изображения. В этом случае, обычно, стремятся ограничить искажение путём его сглаживания (уменьшения пространственной точности). Добиться этого можно путём добавления слагаемого, измеряющего суммарную вариацию adversarial-искажения:

Метрика (10), названная “total variance”, впервые была предложена Mahendran и др. в [70]. Для изображения значение мало, если цвета прилежащих пикселей в нём близки друг к другу, что упрощает возможность воссоздания искажения в реальном мире.

Атакующий может получить сглаженное adversarial-искажение избегая работы с отдельными пикселями. Например, Zhou и др. [71] предложили adversarial-атаку, использующую инфракрасные светодиоды для подсветки лица атакующего, изменяя радиус, яркость и место освещения. При атаке, проиллюстрированной на рисунке 3, освещение лица атакующего незаметно дня окружающих, но воспринимается камерами и позволяет успешно обмануть системы распознавания (авторами [70] в отдельных случаях было получено более 70% успешных обманов)

Рисунок 3 — Направленная adversarial- LED атака [70]

4.2.3 Сложность нахождения adversarial-искажения

Одним из определяющих факторов при совершении атаки на реальные системы может стать скорость, с которой атакующий способен создавать adversarial-искажения. Показательным примером может служить атака на систему [72]. Из-за временных затрат при использовании BIM-метода [6], авторы предпочли использовать более быстрый FGSM-метод [4] генерации adversarial-искажений. Более того, для тестирования параметров атаки требовалась постоянная проверка созданных adversarial-примеров, что, при отсутствии API у атакуемой системы, ограничило возможности автоматизации. Авторы отмечают, что при наличии API предпочтительным способом атаки стало бы обучение модели-близнеца атакуемой системы и проведение против неё white-box атаки.

**5 Известные adversarial-атаки в реальных условиях**

Атаки, описанные в разделе 3.1, наглядно показывают слабые места в развитии моделей глубокого обучения. Тем не менее, встречается мнение (например, в [37][38]), что эти атаки являются не более чем «лабораторным примером», т.к. атакующий имеет прямой доступ к входам модели и эффективность атаки оценивается на стандартном наборе данных. Следовательно, данные атаки не могут быть применены к системам, работающим в реальным мире. Этот раздел нацелен на опровержение данного мнения путём описания атак, проведенных в реальных условиях.

5.1 Атака через камеру мобильного телефона

Впервые переносимость adversarial-атак в условия реального мира продемонстрировали Kurakin и др. в [6]. Они произвели атаки на систему классификации изображений Tensor-Flow Camera [39], использующую модель с архитектурой Inception [40]. Распечатанные на принтере adversarial-изображения (полученные с помощью FGSM, BIM и ILCM) были сфотографированы на камеру сотового телефона и переданы в атакуемую модель. Было показано, что adversarial-искажения сохраняются и в таких условиях, успешно обманывая систему классификации (рисунок 4).

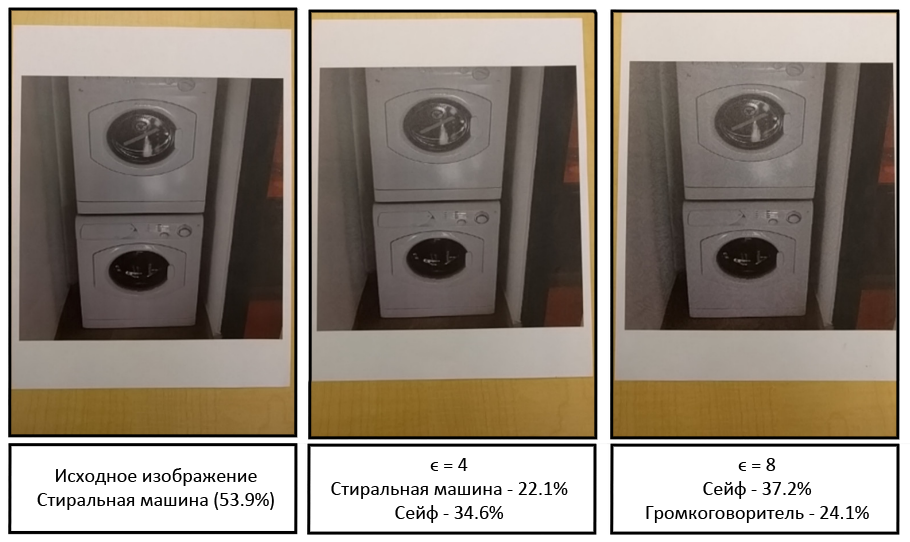


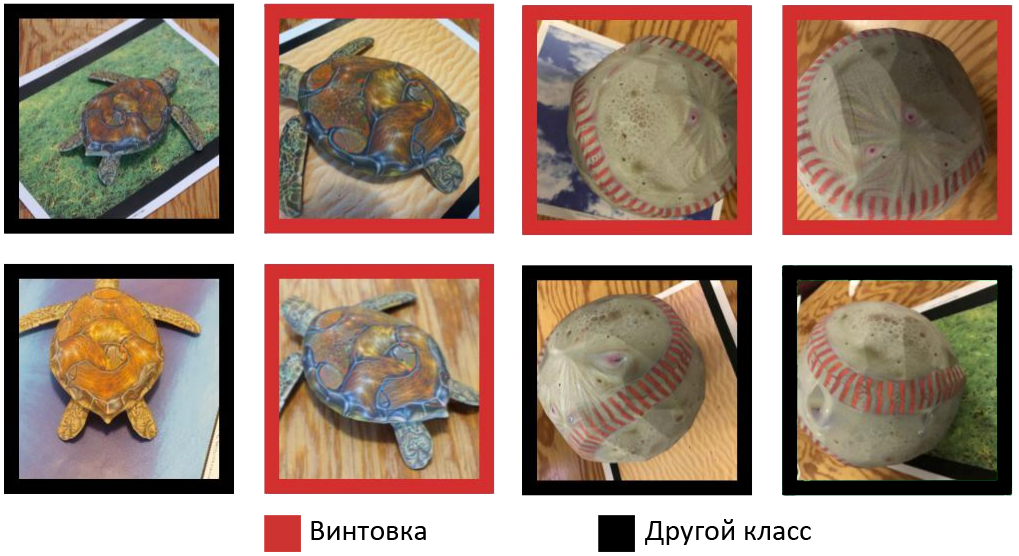
Рисунок 4 — ILCM атака с различными значениями ϵ [6]

5.2 Атака на распознавание дорожных знаков  
 Eykholt и др. [41], на основе C&W атаки [42] и ансамбля моделей, предложенного Lui и др. в [43], успешно сгенерировали adversarial-примеры для системы распознавания дорожных знаков (рисунок 5). Их работа показала возможность создания adversarial-искажений, устойчивых к изменению угла обзора, размера и разрешения изображения. Авторы предложили два способа нанесения adversarial-изменений: 1) полное покрытие знака, при котором атакующий распечатывает полноразмерное adversarial-изображение (рисунок 5а); 2) частичное покрытие знака, при котором adversarial-искажения носят форму наклейки, занимающую только часть дорожного знака и маскируемую под надпись/граффити. (рисунок 5б).

  
а) Изменения, полностью покрывающие знак или его часть. б) Наклейки, замаскированные под граффити.  
Рисунок 5 — Adversarial-искажения для классификатора дорожных знаков LISA [55]

Следует заметить, что данная атака не переносится на современные архитектуры YOLO9000 [44] и FasterRCNN [45], что было показано Lui и др. в [46].   
В похожей работе [47] был разработан метод обучения сети (названной BadNet), в котором модель ошибочно классифицирует дорожный знак при добавлении на него наклейки. При отсутствии наклейки модель показывает результаты, сравнимые с результатами других наилучших моделей. Подчеркнём, что для совершения данной атаки необходимо вмешательство в процесс обучения, то есть совершение её в реальных условиях затруднено.

5.3 Adversarial-3D-модели  
 Athatlye и др. в [48] впервые показали возможность превращения произвольной 3D-модели в adversarial-модель, которая будет являться adversarial-примером при съемке со всех ракурсов. В основе их подхода лежит алгоритм EOT (Expectation Over Transformation), заключающийся в создании adversarial-примера с учётом нескольких изображений из распределения возможных трансформаций (таких как поворот, добавление шума и изменение размера). Такой подход позволяет добиться независимости adversarial-искажения от угла обзора, размера и разрешения изображения. Полученные этим методом 3D-модели, при изначальной правильной классификации, близкой к 100%, изменяют своё предсказание в сторону ложного класса с уверенностью более 80%. Для 3D-моделей, проиллюстрированных на рисунке 6, target-классом был класс «винтовка», уверенность в правильном классе для исходных 3D-моделей близка к 100%. Для серии из 100 снимков с разных углов, предсказания модели по классам равняются: для adversarial-черепахи – винтовка (82%), другое (16%), черепаха (2%); для adversarial-мяча – винтовка (59%), другое (31%), мяч (10%).

  
Рисунок 6 — Классификация 3D-adversarial-моделей

Авторы подчеркивают, что при увеличении размера распределения трансформаций для создания adversarial-3D-модели требуется увеличение вносимого искажения. То есть для создания adversarial-примера, устойчивого к повороту на 30°, требуется внести меньшее искажение, чем для примера, устойчивого к полному повороту и изменению размера. Таким же образом, создание 3D adversarial-примера, обычно, требует внесения большего искажения, чем при создании аналогичного 2D adversarial-изображения.

5.4 Adversarial-очки

Sharif и др. [69] предложили способ создания adversarial-искажений, расположение которых ограничено оправой очков (рисунок 8 – уверенность системы в том, что это один человек – 87.87%). Введя в функцию потерь оптимизатора adversarial-искажений слагаемое NPS (non-printability score), авторы улучшили переносимость созданных искажений в реальные условия. По аналогии с UAP [16], авторы стремились создать adversarial-искажения работающие на множестве лиц. Более чем в 80% случаях им удалось успешно скрыть личность атакующего (ненаправленная атака) и, в некоторых случаях, заставить систему классификации с высокой уверенностью принять атакующего за кого-то другого (направленная атака).

  
Рисунок 8 — Направленная атака на систему распознавания лиц [69]

ЗАКЛЮЧЕНИЕ

В соответствии с классификацией, предложенной в разделе 2, все описанные в разделе 3 методы атак будут отличаться своей направленностью (targeted/non-targeted, т.е. направленная/ненаправленная), информированностью атакующего (black-box/white-box), свойствами искажения (универсальность, используемая норма) и количеством шагов при его получении (одношаговые/итеративные). Все описанные атаки направлены на вызов ошибки 2 рода – изменение верного предсказания модели на ложное.

Таблица 1 — Классификация adversarial-атак раздела 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Атака | Информированность атакующего | Направленность | Количество шагов | Искажение | |
| Универсальность | Норма |
| L-BFGS | White-box | Targeted | Одношаговая | Индивидуальное |  |
| FGSM | White-box | Targeted | Одношаговая | Индивидуальное |  |
| BIM&ILCM | White-box | Non-targeted | Итеративная | Индивидуальное |  |
| JSMA | White-box | Targeted | Итеративная | Индивидуальное |  |
| One-Pixel | Black-box | Non-targeted | Итеративная | Индивидуальное |  |
| C&W | White-box | Targeted | Итеративная | Индивидуальное |  |
| DeepFool | White-box | Non-targeted | Итеративная | Индивидуальное |  |
| UAP | White-box | Non-targeted | Итеративная | Универсальное |  |
| UPSET | Black-box | Targeted | Итеративная | Универсальное |  |
| ANGRY | Black-box | Targeted | Итеративная | Индивидуальное |  |
| Houdini | Black-box | Targeted | Итеративная | Индивидуальное |  |
| ATNs | White-box | Targeted | Итеративная | Индивидуальное |  |

Часто встречается классификация, учитывающая только направленность атаки и информированность атакующего. Такую классификацию удобно проиллюстрировать в виде схемы, изображенной на рисунке 9.

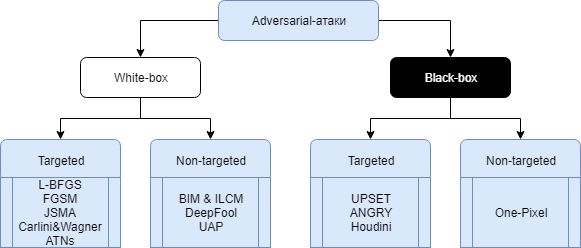


Рисунок 9 — Классификация adversarial-атак раздела 3

Оценка осуществимости атаки в реальных условиях основывается на полном определении соответствия возможным ограничениям, описание которым дано в разделе 4.2. При наложении дополнительных ограничений на adversarial-искажение, как показывают авторы [41][69], существенно возрастает их переносимость в условия реального мира. Особыми факторами, влияющими на осуществимость атаки, являются свойства интерфейсов атакуемой системы (API, защищенность системы). Для однозначного определения осуществимости атаки необходимо определить влияние этих факторов в частном случае, что не позволяет сделать вывод о возможности осуществления атак в общем виде.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. N. Papernot, P. McDaniel, A. Sinha, M. Wellman, “SoK: Towards the science of security and privacy in machine learning”, arXiv:1611.03814, 2016.
2. A. Nguyen, J. Yosinski, J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images”, arXiv:1412.1897, 2015.
3. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, “Intriguing properties of neural networks”, arXiv:1312.6199, 2014.
4. I. J. Goodfellow, J. Shlens, C. Szegedy, “Explaining and Harnessing Adversarial Examples”, arXiv:1412.6572, 2015.
5. A. Kurakin, I. Goodfellow, S. Bengio, “Adversarial Machine Learning at Scale”, arXiv:1611.01236, 2017.
6. A. Kurakin, I. Goodfellow, S. Bengio, “Adversarial examples in the physical world”, arXiv:1607.02533, 2016.
7. T. Miyato, S. Maeda, M. Koyama, S. Ishii, “Virtual Adversarial Training: a Regularization Method for Supervised and Semi-supervised Learning”, arXiv:1704.03976, 2017.
8. I. Sutskever, O. Vinyals, Q. V. Le, “Sequence to sequence learning with neural networks. In Advances in neural information processing systems”, стр. 3104-3112, 2014.
9. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, “The Limitations of Deep Learning in Adversarial Settings”, 2016.
10. J. Su, D. V. Vargas, S. Kouichi, “One pixel attack for fooling deep neural networks”, arXiv:1710.08864, 2017.
11. S. Das, and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art”, журнал “IEEE transactions on evolutionary computation”, номер 15, стр. 4-31, 2011.
12. N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, “Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks”, IEEE Symposium on Security and Privacy, стр. 582-597, 2016.
13. G. Hinton, O. Vinyals, J. Dean, “Distilling the knowledge in a neural network”, NIPS 2014, arXiv:1503.02531, 2014.
14. P. Chen, H. Zhang, Y. Sharma, J. Yi, C. Hsieh, “ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models”, 10th ACM Workshop on Artificial Intelligence and Security, 2017.
15. S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, “DeepFool: a simple and accurate method to fool deep neural networks”, IEEE Conference on Computer Vision and Pattern Recognition, 2574-2582, 2016.
16. S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, “Universal adversarial perturbations”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
17. V. Khrulkov, I. Oseledets, “Art of singular vectors and universal adversarial perturbations”, arXiv:1709.03582, 2017.
18. K. R. Mopuri, U. Garg, R. V. Babu, “Fast Feature Fool: A data independent approach to universal adversarial perturbations”, arXiv:1707.05572, 2017.
19. S. Sarkar, A. Bansal, U. Mahbub, R. Chellappa, “UPSET and ANGRI: Breaking High Performance Image Classifiers”, arXiv:1707.01159, 2017.
20. A. Krizhevsky, “Learning multiple layers of features from tiny image”, 2009.
21. M. Cisse, Y. Adi, N. Neverova, J. Keshet, “Houdini: Fooling deep structured prediction models”, arXiv:1707.05373, 2017.
22. D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, “Deep speech 2: End-to-end speech recognition in English and Mandarin”, arXiv:1512.02595, 2015.
23. S. Baluja, I. Fischer, “Adversarial Transformation Networks: Learning to Generate Adversarial Examples”, arXiv:1703.09387, 2017.
24. J. Hayes, G. Danezis, “Machine Learning as an Adversarial Service: Learning Black-Box Adversarial Examples”, arXiv:1708.05207, 2017.
25. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, “Backpropagation applied to handwritten zip code recognition”, журнал “Neural computation”, номер 4, стр. 541-551, 1989.
26. S. Sabour, Y. Cao, F. Faghri, D. J. Fleet, “Adversarial manipulation of deep representations”, arXiv:1511.05122, 2015.
27. N. Papernot, Patrick McDaniel, Ian Goodfellow, “Transferability in machine learning: from phenomena to black-box attacks using adversarial samples”, arXiv:1605.07277, 2016.
28. N. Narodytska, S. P. Kasiviswanathan, “Simple Black-Box Adversarial Perturbations for Deep Networks”, arXiv:1612.06299, 2016.
29. Y. Liu, W. Zhang, S. Li, N. Yu, “Enhanced Attacks on Defensively Distilled Deep Neural Networks”, arXiv:1711.05934, 2017.
30. S. J. Oh, M. Fritz, B. Schiele, “Adversarial Image Perturbation for Privacy Protection – A Game Theory Perspective”, arXiv:1703.09471, 2017.
31. H. Hosseini, B. Xiao, M. Jaiswal, R. Poovendran, “On the Limitation of Convolutional Neural Networks in Recognizing Negative Images”, arXiv:1703.06857, 2017.
32. C. Kanbak, SS. Moosavi-Dezfooli, P. Frossard, “Geometric robustness of deep networks: analysis and improvement”, arXiv:1711.09115, 2017.
33. Y. Dong, F. Liao, T. Pang, H. Su, X. Hu, J. Li, J. Zhu, “Boosting Adversarial Attacks with Momentum”, arXiv:1710.06081, 2017.
34. N. Carlini, G. Katz, C. Barrett, D. L. Dill, “Ground-Truth Adversarial Examples”, arXiv:1709.10207, 2017.
35. A. Rozsa, E. M. Rudd, T. E. Boult, “Adversarial Diversity and Hard Positive Generation”, arXiv:1605.01775, 2016.
36. Y. Yoo, S. Park, J. Choi, S. Yun, N. Kwak, “Butterfly Effect: Bidirectional Control of Classification Performance by Small Additive Perturbation”, arXiv:1711.09681, 2017.
37. J. Lu, H. Sibai, E. Fabry, D. Forsyth, “No need to worry about adversarial examples in object detection in autonomous vehicles”, arXiv:1707.03501, 2017.
38. A. Graese, A. Rozsa, T. E. Boult, “Assessing Threat of Adversarial Examples on Deep Neural Networks”, International Conference on Machine Learning and Applications, стр. 69-74, 2016.
39. Objects Detection Machine Learning TensorFlow Demo, https://play.google.com/store/apps/details?id=org.tensorflow.detect, последнее обращение – июнь 2019.
40. C. Szegedy, V. Vincent, S. Ioffe, J. Shlens, Z. Wojna, “Rethinking the inception architecture for computer vision”, IEEE Conference on Computer Vision and Pattern Recognition, 2818-282, 2016.
41. I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, D. Song, “Robust Physical-World Attacks on Deep Learning Models”, arXiv:1707.08945, 2017.
42. N. Carlini, D. Wagner, “Towards Evaluating the Robustness of Neural Networks”, arXiv:1608.04644, 2016.
43. Y. Liu, X. Chen, C. Liu, D. Song, “Delving into Transferable Adversarial Examples and Black-box Attacks”, arXiv:1611.02770, 2017.
44. J. Redmon, A. Farhadi. “Yolo9000: Better, Faster, Stronger”, arXiv:1612.08242, 2016.
45. S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards realtime object detection with region proposal networks”, журнал “Advances in neural information processing systems”, стр. 91-99, 2015.
46. J. Lu, H. Sibai, E. Fabry, D. Forsyth, “Standard detectors aren’t (currently) fooled by physical adversarial stop signs”, arXiv:1710.03337, 2017.
47. T. Gu, B. Dolan-Gavitt, S. Garg, “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain”, arXiv:1708.06733, 2017.
48. A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, “Synthesizing Robust Adversarial Examples”, arXiv:1707.07397, 2017.
49. E. Ackerman, “How Drive.ai is mastering autonomous driving with deep learning”, https://spectrum. ieee.org/cars-that-think/transportation/self-driving/how-driveai-is-mastering-autonomous-driving-with-deep-learning, последнее обращение – июнь 2019.
50. “Deep Learning for Self-Driving Cars”, https://class-central.com/mooc/8132/6-s094-deep-learning-for-self-driving-cars, последнее обращение – июнь 2019.
51. M. Volodymyr, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, “Human-level control through deep reinforcement learning”, журнал “Nature”, номер 518/7540, стр. 529-533, 2015.
52. A. Giusti, J. Guzzi, D. C. Ciresan, F. He, J. P. Rodriguez, F. Fontana, M. Faessler, “A machine learning approach to visual perception of forest trails for mobile robots”, “IEEE Robotics and Automation Letters”, номер 1/2, стр. 661 - 667, 2016.
53. C. Middlehurst, “China unveils world’s first facial recognition ATM”, https://www.telegraph.co.uk/news/worldnews/asia/china/11643314/China-unveils-worlds-first-facial-recognition-ATM.html, последнее обращение – июнь 2019.
54. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”, журнал “Signal Processing”, номер 29/6, стр. 82-97, 2012.
55. A. Mogelmose, M. M. Trivedi, T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey”, IEEE Transaction on Intelligent Transportation Systems, номер 13/4, стр. 1484-1497, 2012.
56. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”, IEEE Conference on Computer Vision and Pattern Recognition, стр. 248-255, 2009.
57. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-Based learning applied to document recognition”, журнал “Proceedings of the IEEE”, 1998.
58. K. Simonyan, A. Zisserman, “Very deep CNN for large-scale image recognition”, arXiv: 1409.1556, 2015.
59. A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet classification with Deep CNNs”, 2012.
60. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, “Caffe: Convolutional Architecture for Fast Feature Embedding”, arXiv: 1408.5093, 2014.
61. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, arXiv: 1512.03385, 2015.
62. B. Biggio, B. Nelson, and P. Laskov, “Poisoning attacks against support vector machines”, arXiv:1206.6389, 2012.
63. F. Roli, B. Biggio, and G. Fumera, “Pattern recognition systems under attack”, 2013.
64. J. Lu, T. Issaranon, D. Forsyth, “SafetyNet: Detecting and Rejecting Adversarial Examples Robustly”, arXiv:1704.00103, 2017.
65. W. Xu, D. Evans, Y. Qi, “Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks”, arXiv:1704.01155, 2017.
66. W. Xu, D. Evans, and Y. Qi, “Feature Squeezing Mitigates and Detects Carlini/Wagner Adversarial Examples”, arXiv:1705.10686, 2017.
67. X. Li, F. Li, “Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics”, International Conference on Computer Vision, 2017.
68. Dongyu Meng, Hao Chen, “MagNet: a Two-Pronged Defense against Adversarial Examples”, ACM Conference on Computer and Communications Security, 2017.
69. M. Sharif, S. Bhagavatula, L. Bauer, M. Reiter “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition”, ACM Conference on Computer and Communications Security, стр. 1528–1540, 2016.
70. A. Mahendran, A. Vedaldi, “Understanding Deep Image Representations by Inverting Them”, arXiv: 1412.0035, 2014.
71. Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, Kehuan Zhang, “Invisible Mask: Practical Attacks on Face Recognition with Infrared”, arXiv: 1803.04683, 2018.
72. Илья Сергеев, “Как мы боремся с копированием контента”, https://habr.com/ru/company/avito/blog/452142/, последнее обращение – июнь 2019.
73. D. P. Kingma, J. Ba, “Adam: A Method for Stochastic Optimization”, arXiv:1412.6980, 2014.