УДК 519.6 DOI: 10.18698/0236-3933-2021-1-102-119

КАК ОБМАНУТЬ НЕЙРОННУЮ СЕТЬ? СИНТЕЗ ШУМА ДЛЯ УМЕНЬШЕНИЯ ТОЧНОСТИ НЕЙРОСЕТЕВОЙ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

А.П. Карпенко apkarpenko@bmstu.ru
В.А. Овчинников ovchinnikov.vadim.a@gmail.com

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Цель исследования — разработка алгоритмического и программного обеспечения для синтеза шума с намерением совершения атак на нейронные сети глубокого обучения, предназначенные для классификации изображений. Приведены результаты анализа методов проведения атак на такие нейронные сети. Задача синтеза «атакующего» шума сформулирована как задача многомерной условной оптимизации. Основные особенности предложенного алгоритма синтеза атакующего шума заключаются в следующем: ограничения на шум учитываются с помощью функции clip; в качестве критериев эффективности атакующего шума используются рейтинги top-1, top-5 ошибок классификации; для обучения нейронных сетей применяются алгоритмы обратного распространения ошибки и градиентного спуска Адама; указанная задача оптимизации решается методом стохастического градиентного спуска; в процессе обучения нейронных сетей используется техника аугментации. Разработанное программное обеспечение работает под управлением операционных систем *Ubuntu 18.04* и *CentOS 7*, написано на языке программирования *Python* с использованием фреймворка динамического дифференцирования графа вычислений *Pytorch*. Среда разработки *Visual Studio Code*. Для ускорения вычислений использованы графический процессор *Nvidia TITAN XP* и технология *CUDA*. Приведены результаты широкого вычислительного эксперимента по синтезу неуниверсального и универсального атакующих шумов для восьми глубоких нейронных сетей. Показано, что предложенный атакующий алгоритм может увеличивать ошибку нейронной сети в 8 раз

Ключевые слова

Глубокая нейронная сеть, классификация изображений, синтез атакующего шума, графические ускорители

Поступила 11.03.2020
Принята 25.06.2020

Введение. В настоящее время с использованием нейронных сетей глубокого обучения успешно решают такие задачи, как распознавание речи [1], ее генерация [2], классификация изображений [3] и др. Аппаратную основу успеха решения перечисленных задач создают высокопроизводительные графические процессорные устройства (GPU).

Развитие глубоких нейронных сетей как одной из технологий машинного обучения сделало актуальной проблему обеспечения безопасности информационных систем, построенных на их основе. Область исследования атак на нейронные сети довольно нова — первые серьезные работы в этой области опубликованы в 2013 г. За прошедшие годы эти исследования получили развитие, и в настоящее время существует значительное число способов «обмана» нейросетевых систем обработки данных. Вместе с развитием методов атак на нейронные сети развиваются методы защиты от этих атак. Методика повышения устойчивости нейронных сетей к атакам основана на использовании атакованных данных для обучения этих сетей — *обучение атаками* (*adversarial training*). Такое обучение позволяет не только уменьшить восприимчивость нейронных сетей к атакам, но и повысить точность этих сетей.

Значительное число работ посвящено исследованию эффективности атак на глубокие нейронные сети, которые решают конкретные задачи классификации изображений. Так, в [4] исследована глубокая сеть, предназначенная для распознавания лиц. Авторы предложили алгоритм синтеза атакующего шума, который с высокой вероятностью позволяет изменять распознаваемый пол человека [4]. Эффективность атаки на нейронную сеть, распознающую дорожные знаки, исследована в [5]. Авторы синтезировали атакованный дорожный знак и наклеивали его поверх настоящего знака. Показана высокая эффективность атак для нескольких современных нейросетевых систем распознавания дорожных знаков.

Постановка задачи классификации изображений. Суть задачи классификации изображений состоит в отнесении изображения к *ожидаемому* классу, который может определять некоторые события или объект на изображении. Пример задачи классификации изображений приведен в работе [6], в которой предложена сверточная нейронная сеть для классификации цифр, обученная на наборе данных *MNIST* [7]. Этот набор содержит 60 000 изображений рукописных цифр от 0 до 9, так что каждое изображение относится к одному из 10 классов, имя которого соответствует изображенной цифре.

Задачу обучения классифицирующей нейронной сети сформулируем следующим образом. Имеется совокупность (входных) изображений

$\mathbf{I} = \{I_1, ..., I_{|\mathbf{I}|}\}$ и набор имен классов $C = \{c_1, ..., c_{|C|}\}$, которым принадлежат эти изображения. Отображение $\mathbf{I} \to C$ известно для изображений *обучающей выборки* $\{(I_1^l, c_1^l), ..., (I_m^l, c_m^l)\}$, где m — число изображений в выборке; $I_i^l \in \mathbf{I}$; $c_i^l \in C$. Требуется построить *классифицирующий нейросетевой алгоритм* $A_{NN}(I, W)$, который может правильно классифицировать произвольное изображение $I \in \mathbf{I}$. Здесь W — вектор параметров (в частности, весов) нейронной сети NN.

Для отыскания алгоритма $A_{NN}(I, W)$ решают оптимизационную задачу вида

$$E\left(A_{NN}(I, W),\, C\right) = \frac{1}{m}\left(A_{NN}\left(I_i^l, W\right),\, c_i^l\right) \to \min_W,$$

где $E(\cdot)$ — критерий эффективности классификации, в качестве которого может выступать, например, функция потерь [8].

Обзор типов и методов атак на классифицирующие нейронные сети. Атаку на нейронную сеть определяют как искажения входного изображения, которое не может быть детектировано экспертом, но которое может привести к ошибке классификации нейронной сетью этого изображения.

Пусть I_o — исходное (оригинальное) изображение, I_p — искаженное изображение (полученное в результате атаки на нейронную сеть). Разность $P = (p_i, i[1:|P|]) = I_o - I_p$ принято называть (атакующим) шумом (*adversarial perturbation*). Выделяют следующие типы атак на нейронные сети: открытые и закрытые; направленные и ненаправленные; универсальные и неуниверсальные.

В случае *открытых* (*white box*) атак для генерации шума необходимо иметь доступ к нейронной сети, на которую совершается атака. На первый взгляд такая ситуация маловероятна и открытые атаки не представляют серьезной опасности. Однако это не так. В настоящее время широко известны наиболее эффективные нейронные сети — фавориты, предназначенные для решения различных задач классификации и находящиеся в открытом доступе: для распознавания речи — нейронные сети *Tacotron* [1] и *WaveNet* [9]; для классификации изображений — *Inceptionv4* [10] и т. д. Поэтому многие компании используют эти сети в коммерческих продуктах. Для *закрытых* (*black box*) методов необходимо знать только результат классификации нейронной сетью.

Направленные атаки могут изменить класс изображения на необходимый атакующему, а *ненаправленные* атаки — на любой класс, отлич-

ный от исходного. Как правило, алгоритмы, предназначенные для осуществления ненаправленных атак, обладают невысокой вычислительной сложностью. Синтезируемый этими алгоритмами шум имеет малые по абсолютной величине значения. Алгоритмы для направленных атак имеют более высокую вычислительную сложность и наиболее широко применяются для совершения реальных атак [4, 10, 11].

Неуниверсальные атаки предполагают поиск (уникального) шума для каждого конкретного изображения. Поскольку поиск такого шума требует высоких вычислительных затрат, актуальной задачей является разработка алгоритмов для *универсальных* атак [12, 13].

Метод синтеза атакующего шума для нейросетевого классификатора изображений должен удовлетворять основным требованиям:

– легко встраиваться в процесс обучения классификатора;

– иметь время выполнения, значительно меньшее времени одной эпохи обучения классификатора;

– синтезировать шум, незаметный или слабозаметный для глаза человека;

– иметь высокие показатели эффективности атак.

Метод синтеза атакующего шума P, незаметного для глаза человека, но способного «обмануть» современные высокоточные нейронные сети, впервые предложен в [14]. Метод основан на решении задачи оптимизации

$$\lambda \|P\|_\infty + E(A_{NN}(I_o + P, W), c_f) \to \min_P, \tag{1}$$

где λ — весовой множитель; $\|P\|_\infty = \max\limits_{i \in [1:|P|]} abs(p_i)$ — бесконечная норма шума; $E(\cdot)$ — подлежащий минимизации критерий эффективности классификации; c_f — один из неверных классов изображений. Метод продуцирует направленную атаку, позволяющую изменить класс исходного изображения на класс c_f, необходимый атакующему. Для решения задачи (1) авторы используют известный алгоритм *L-BFGS* [20].

Эффективный метод (*fast gradient sign method*) генерации атакующего шума для изображений разработан в [12]. Шум по этому методу определяют по формуле

$$P = \varepsilon \operatorname{sign}\left(\nabla\left(-E\left(A_{NN}(I_o, W), c\right)\right)\right), \tag{2}$$

где ε — малая величина, достаточная для того, чтобы шум не был заметен для эксперта. Модификация метода, предложенная в [15], в качестве класса c использует ложный класс c_f, превращая тем самым метод

в направленный. В формуле (2) в [16] предложено использовать не знак градиента, а градиент, нормализованный с помощью норм $\|\cdot\|_2$ или $\|\cdot\|_\infty$.

Итерационный метод синтеза атакованного изображения в [11] предложено определять по формуле

$$I_p^{t+1} = \text{clip}\left(I_p^t + \varepsilon\nabla\left(-E\left(A_{NN}\left(I_o, W\right), c\right)\right)\right), \tag{3}$$

где clip — функция, которая ограничивает значения аргумента интервалом $[0;1]$; t — номер итерации. Метод легко модифицируется для генерации направленных атак путем замены класса c в формуле (3) ложным классом c_f. Развитием метода [11] можно полагать метод *DeepFool* [17]. Этот метод способен синтезировать атакующий шум, меньший по абсолютному значению, чем шум, найденный быстрым методом знаков градиентов [12], и обеспечивать примерно такую же ошибку нейронной сети, как и в последнем методе.

Рассмотренные методы атак используют для генерации атакующего шума все пиксели исходного изображения. Использовать для атаки только один «уязвимый» пиксель изображения предложено в [18]. В целях поиска этого пикселя применяют метод дифференциальной эволюции [19], а не градиентные методы, как в представленных выше публикациях. Метод является закрытым, направленным и неуниверсальным.

Рассмотренные методы предназначены для синтеза неуниверсального атакующего шума. Метод генерации универсального шума предложен в [13]. При изменении интенсивности каждого пикселя изображения не более чем на 4 % метод обеспечивает примерно 80 % ложных срабатываний лучших нейросетевых классификаторов изображений. Для обучения шума использовалось от 1 до 5 тыс. изображений (что значительно меньше, чем в наборе данных *Imagenet* [20]).

Постановка задачи синтеза атакующего шума и предлагаемый метод решения задачи. С учетом введенных выше обозначений задачу синтеза атакующего шума для нейронной сети NN формулируем следующим образом. Требуется найти шум $P \in [P^-; P^+]$, который максимизирует критерий эффективности классификации $E(P)$. Здесь P^-, P^+ — векторы, определяющие нижнюю и верхнюю границы допустимых значений соответствующих компонентов шума P; $\left|P^-\right| = \left|P^+\right| = \left|P\right|$. Задачу поиска такого шума ставим как задачу глобальной условной оптимизации вида

$$E(P) = \frac{1}{m}\sum_{i=1}^{m} E\left(A_{NN}\left(I_i^l + P, W\right), c_i^l\right) \to \max_{P \in [P^-;\, P^+]}, \tag{4}$$

где A_{NN} — алгоритм классификации, реализуемый нейронной сетью NN. Нейронную сеть NN полагаем предобученной, так что вектор W весов этой сети фиксирован.

Вектору искомых шумов P ставим в соответствие входной слой *расширенной* нейронной сети ENN. Реализуемый этой сетью алгоритм классификации обозначаем как $\overline{A}_{ENN}$. В этих обозначениях задача (4) приобретает вид

$$E(P) = \frac{1}{m}\sum_{i=1}^{m} E\left(\overline{A}_{ENN}\left(I_i^l,\ P,\ W\right),\ c_i^l\right) \to \max_{P\in[P^-;P^+]}, \tag{5}$$

т. е. становится задачей обучения нейронной сети ENN.

По общим правилам для учета ограничений на компоненты вектора шума P в задаче (5) можно использовать методы барьерных или штрафных функций. Однако такой подход требует слишком высоких вычислительных затрат. Поэтому применяем простую схему ограничения шума на основе формулы

$$P = \mathrm{clip}(P,\ P^-,\ P^+).$$

В качестве критерия эффективности $E(P)$ атакующего шума P используем классические рейтинги top-1, top-5 ошибки классификации, имеющие в таком случае смысл оценки вероятности принадлежности исходного изображения первому наиболее вероятному классу, предсказанному сетью NN, и аналогичным пяти первым наиболее вероятным классам соответственно.

Для обучения рассматриваемых расширенных нейронных сетей применяем связку алгоритмов обратного распространения ошибки [7] и стохастического градиентного спуска Адама [21, 22]. Условие окончания обучения нейронных сетей — достижения заданного числа эпох обучения M, которое широко применяют в процессе обучения глубоких нейронных сетей.

Из набора *Imagenet* выделено подмножество, включающее в себя 50 000 изображений, разделенных на три части:

1) тренировочная выборка (4000 изображений);

2) валидационная выборка (1000 изображений);

3) тестовая выборка (45 000 изображений).

Отметим, что набор данных *Imagenet* [20] содержит 14 млн изображений 1000 классов.

Тренировочную выборку используем для обучения нейронных сетей ENN, так что размер обучающей выборки равен $m = 4000$. Применяем технику аугментации, которая в общем случае позволяет искусственно увеличить объем выборки путем вырезания частей изображения, сжатия,

растяжения и поворотов. Используем аугментацию на основе следующего набора операций над изображениями: растяжение изображения до 256×256 пикселей; вырезание из полученного изображения случайного кадра размером 224×224 пикселей; случайный поворот изображения в его плоскости. Таким образом, в формуле (5) в действительности используются не изображения I_i^l, а их случайные трансформации с использованием указанных операций над изображениями.

В целях обеспечения статистической достоверности результатов исследования используем метод мультистарта — каждую нейронную сеть ENN обучаем несколько раз, исходя из различных случайных начальных значений шума.

Валидационная выборка применяется для обнаружения эффекта переобучения рассматриваемых расширенных нейронных сетей. Тестовая выборка предназначена для оценки эффективности предложенного алгоритма генерации шума.

Программное обеспечение. Разработанное программное обеспечение *WorstNoise* реализует схему исследования. Программа, написанная на языке программирования *Python* с использованием фреймворка динамического дифференцирования графа вычислений *Pytorch*, работает под управлением операционных систем *Ubuntu 18.04* и *CentOS 7*. Среда разработки *Visual Studio Code*. Для ускорения вычислений использован графический процессор *Nvidia TITAN XP* и технология *CUDA*.

Pytorch — открытый фреймворк компании *Facebook*, который реализует алгоритм автоматического дифференцирования динамического графа вычислений с использованием алгоритма обратного распространения ошибки. Кроме того, библиотеки *Pytorch* содержат программные реализации современных методов оптимизации, основанных на вычислении градиента оптимизируемой функции. В этом фреймворке также реализованы базовые механизмы аугментации и загрузки данных.

Программное обеспечение *WorstNoise* включает в себя следующие модули: модуль загрузки обученных нейронных сетей из библиотеки *torchvison*; модуль чтения и аугментации данных; модуль синтеза атакующего шума; модуль, реализующий вычисление значений рейтингов, полученных на этапах валидации и тестирования. Каждый модуль содержит основные и вспомогательные функции.

Вычислительный эксперимент и обсуждение его результатов. Для тестирования эффективности разработанного алгоритмического и программного обеспечения используем набор $\{NN_k, k \in [1:8]\}$, состо-

ящий из шести *основных* нейронных сетей $NN_1 - NN_6$ и двух *дополнительных* нейронных сетей NN_7, NN_8:

1) *alexnet* (NN_1) — улучшенный вариант сети *LeNet* [11] (около 60 млн параметров) [3];

2) *vgg19* (NN_2) — 19 слоев, примерно 144 млн параметров [23];

3) *resnet152* (NN_3) — 152 слоя, 60 млн параметров [24];

4) *densenet* (NN_4) — 20 млн параметров [25];

5) *googlenet* (NN_5) — 11 млн параметров [26];

6) *mobilenet* (NN_6) — сеть для использования на мобильных устройствах (около 6,9 млн параметров) [27];

7) *vgg11* (NN_7) — 11 слоев [23];

8) *resnet18* (NN_8) — 18 слоев [24].

Указанные глубокие нейронные сети отличаются высокой точностью и приемлемой вычислительной сложностью обучения. Значения ошибок для основных сетей приведены в табл. 1 [28].

Таблица 1

Значения ошибок основных неатакованных глубоких нейронных сетей

Рейтинг	Нейронная сеть					
	alexnet	*vgg19*	*resnet152*	*densenet*	*googlenet*	*mobilenet*
top-1 (%)	43,4	27,6	21,7	22,4	30,2	28,1
top-5 (%)	20,9	9,1	5,9	6,2	10,5	9,7

Вычислительный эксперимент выполнен при следующих значениях свободных параметров атакующего алгоритма:

$$P^- = (-0,05)^{|P|},\ P^+ = (0,05)^{|P|},$$

$$l = 0,01,\ \beta_1 = 0,9,\ \beta_2 = 0,999,\ \varepsilon = 10^{-8},\ M = 20,\ |P| = 224 \times 224 \times 3.$$

Здесь $l, \beta_1, \beta_2, \varepsilon$ — величины, представляющие собой параметры алгоритма Адама; размерность $|P|$ вектора шума определяется размером используемых изображений (224 × 224 пикселей) и цветовой моделью RGB. Указанные значения параметров P^-, P^+ означают, что допускаются 5%-ные изменения значений каждой составляющей RGB каждого пикселя изображения.

Процесс стагнации целевой функции E в ходе решения задачи (4) приведен на рис. 1, *а*. Согласно рисунку, значительные среднеквадратичные отклонения имеют место только для сети *resnet152* и только

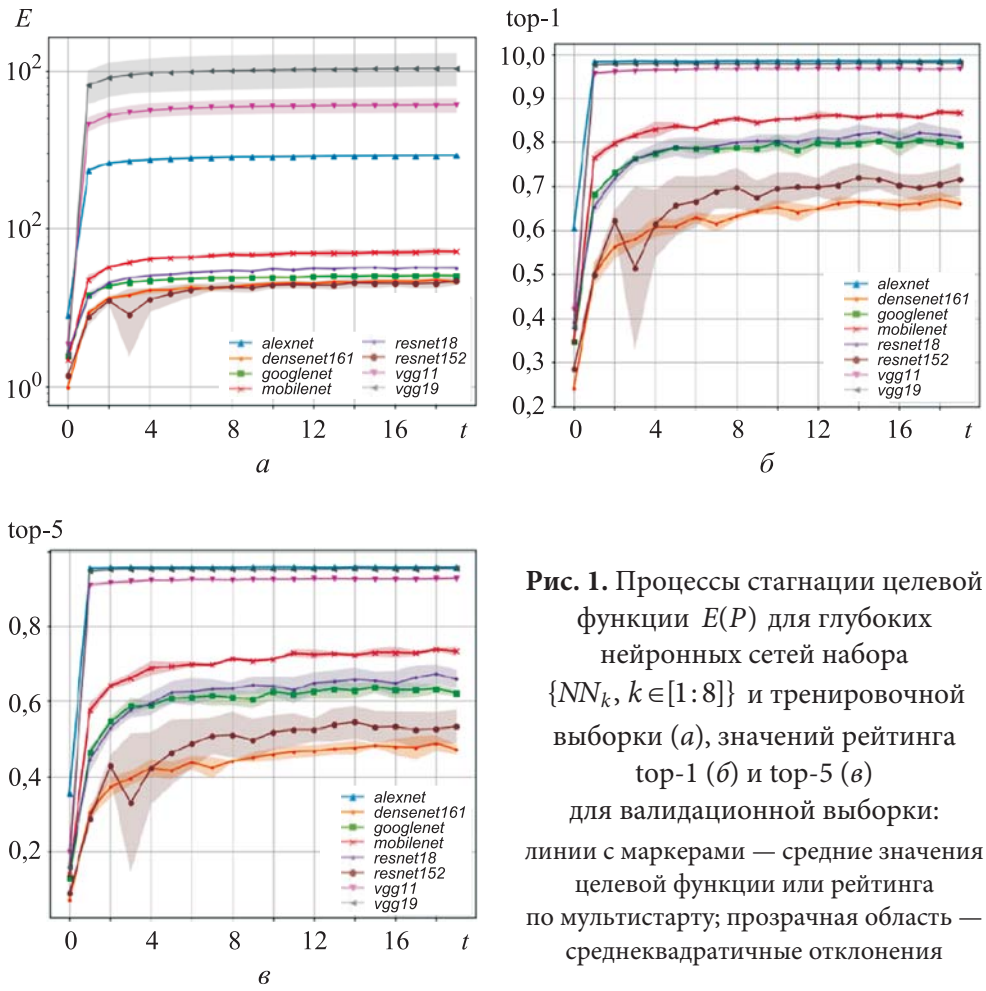


Рис. 1. Процессы стагнации целевой функции $E(P)$ для глубоких нейронных сетей набора $\{NN_k, k \in [1:8]\}$ и тренировочной выборки (*а*), значений рейтинга top-1 (*б*) и top-5 (*в*) для валидационной выборки: линии с маркерами — средние значения целевой функции или рейтинга по мультистарту; прозрачная область — среднеквадратичные отклонения

на начальных итерациях t. Для всех рассматриваемых нейронных сетей на пятой итерации удалось достичь высоких значений целевой функции, которые в результате дальнейших итераций улучшились незначительно. Таким образом, предложенный алгоритм решения задачи (4) обеспечивает высокую скорость сходимости и поэтому в плане вычислительных затрат является легковесным по сравнению с затратами на одну эпоху обучения рассматриваемых нейросетевых классификаторов.

Характер изменения рейтинга top-1 на валидационном наборе данных показан на рис. 1, *б*. Как и на рис. 1, *а*, имеет место большое среднеквадратичное отклонение точности сети *resnet* на начальных итерациях. С одной стороны, заметен значительный разрыв между эффективностями кластеров сетей *vgg*, *alexnet*, с другой, с остальными сетями. Это явление объясняется близостью архитектур этих сетей. Синтезированный атакующий шум поз-

волил повысить значение рейтинга top-1 для самой устойчивой сети *densenet* с 22,3 до 66,9 %, что сделало эту сеть хуже сети *alexnet*.

Характер изменения значений рейтинга top-5 на валидационном наборе данных показан на рис. 1, *в*. Из рисунка следует, что нейронная сеть *densenet* продолжает доминировать. Синтезированный атакующий шум позволил повысить значения рейтинга с 6,2 до 48,8 %.

Представленные результаты исследования показывают, что предложенный атакующий алгоритм способен увеличивать ошибку нейронной сети в 8 раз. Однако эти результаты получены на валидационной выборке малого объема, которая не может быть классифицирована как репрезентативная.

Интегральные результаты, полученные на тестовой выборке, объем которой в 9 раз больше суммарного объема тренировочной и валидационной выборок, приведены в табл. 2. Полученные для тестовой и валидационной выборок результаты близки. Это свидетельствует о том, что предложенный атакующий алгоритм может значительно снизить точность глубоких нейросетевых классификаторов даже при использовании обучающей выборки малого объема.

Таблица 2

Значения ошибок основных атакованных глубоких нейронных сетей для тестовой выборки

Рейтинг	Нейронная сеть					
	alexnet	*vgg19*	*resnet152*	*densenet*	*googlenet*	*mobilenet*
top-1 (%)	98,7	98,3	71,9	66,9	80,0	87,4
top-5 (%)	96,2	95,7	54,4	48,9	63,6	74,4

Усредненные оценки эффективности предложенного атакующего алгоритма как генератора универсального алгоритма приведены в табл. 3. Представлены усредненные оценки эффективности шума, синтезированного для этой атакуемой нейронной сети, с позиции других сетей.

Таблица 3

Усредненные оценки эффективности атакующего шума как генератора универсального шума для тестовых выборок

Атакуемая сеть	Атакующая сеть					
	alexnet	*vgg19*	*resnet152*	*densenet*	*googlenet*	*mobilenet*
	Рейтинг top-1					
alexnet	98,7	79,4	36,5	35,8	48,4	56,1

Окончание табл. 3

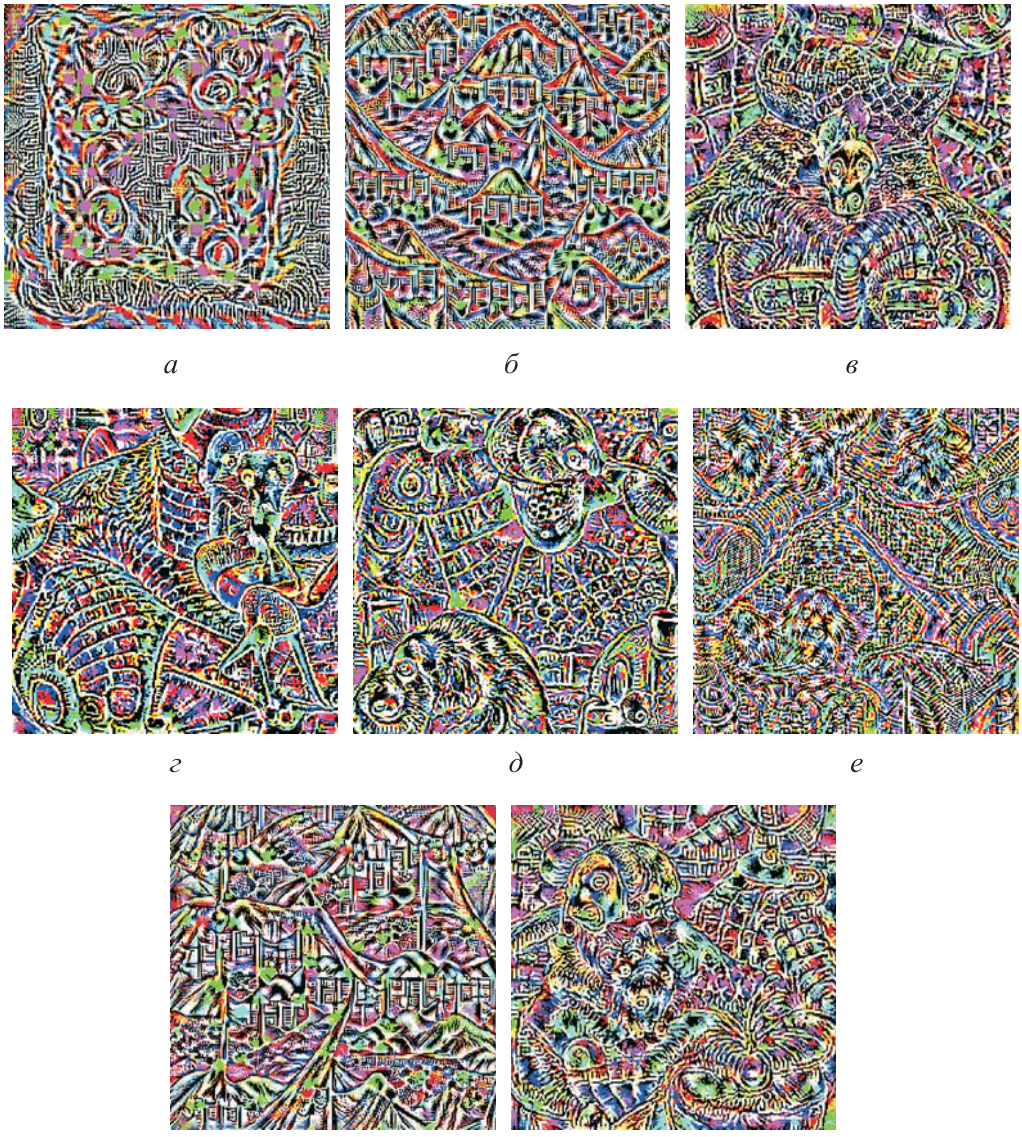
Атакуемая сеть	Атакующая сеть					
	alexnet	*vgg19*	*resnet152*	*densenet*	*googlenet*	*mobilenet*
Рейтинг top-1						
vgg19	81,1	98,3	34,9	35,0	45,5	54,0
resnet152	78,7	74,7	71,9	43,9	52,3	60,4
densenet	79,8	77,6	43,7	66,9	51,1	59,9
googlenet	78,5	74,3	40,8	38,6	80,0	57,9
mobilenet	75,5	67,9	32,9	34,0	43,5	87,4
Рейтинг top-5						
alexnet	96,2	61,6	15,8	15,4	24,8	32,2
vgg19	61,7	95,7	14,5	14,8	22,4	29,8
resnet152	58,1	54,9	54,4	23,7	29,0	37,8
densenet	60,1	59,2	22,6	48,9	27,9	36,7
googlenet	57,8	54,8	20,1	18,4	63,6	35,0
mobilenet	53,5	45,7	13,3	14,3	20,6	74,4

С позиции универсальности бесспорными лидерами являются атакующие шумы, сгенерированные для сетей *resnet*, *densenet*. Эти шумы смогли увеличить ошибку всех рассматриваемых сетей до 43 %. В среднем предложенный алгоритм синтеза атакующего шума увеличил значение рейтинга top-1 этих сетей в 2 раза. Таблица показывает «зеркальность» ошибок таких сетей, как *alexnet*/*vgg* и *resnet*/*densenet*. Это обстоятельство свидетельствует о близости в некотором смысле архитектур указанных сетей, а также, возможно, о похожем способе преобразования изображений в этих сетях во внутреннее представление.

С позиции рейтинга top-5 лидерами являются те же сети, что и с позиции рейтинга top-1. Синтезированные для этих сетей шумы смогли увеличить ошибки остальных сетей до 22 %, т. е. примерно в 3 раза выше, чем у неатакованных классификаторов.

Полученные атакующие шумы приведены на рис. 2. Отметим, что в каждом представленном шуме наблюдается некоторая структура, причем нейронные сети, имеющие близкие архитектуры (*vgg11*/*vgg19* и *resnet18*/*resnet152*), порождают схожие структуры.

Заключение. Предложенный алгоритм синтеза атакующего шума обладает универсальностью, открытостью, ненаправленностью, малым числом существенных свободных параметров (только минимальное и мак-



а *б* *в*

г *д* *е*

ж *з*

Рис. 2. Атакующие шумы, синтезированные для основных и вспомогательных глубоких нейронных сетей $\{NN_k,\ k \in [1:8]\}$:

а — alexnet; *б — vgg19*; *в — resnet152*; *г — densenet*; *д — googlenet*; *е — mobilenet*; *ж — vgg11*; *з — resnet18*

симальное допустимое значение синтезируемого шума), невысокой вычислительной сложностью, малой заметностью синтезированного шума для глаза человека.

Алгоритм показал эффективность как для атакуемой сети, так и для других сетей (универсальность атакующего шума). Для ненаправленных

атак установлена следующая важная особенность синтезированного атакующего шума: наличие в обучающей выборке изображений близких классов незначительно снижает эффективность атак (рейтинг top-5 незначительно уменьшается по сравнению с рейтингом top-1).

Исследование, близкое к проведенному, выполнено в [13] для набора нейронных сетей глубокого обучения *vgg-f*, *caffenet*, *googlenet*, *vgg16*, *vgg19*, *resnet152* (включает в себя рассмотренные выше нейронные сети *vgg19*, *resnet152*, *googlenet*). Найденные атакующие шумы близки по эффективности к шумам, полученным в настоящей работе. Однако авторы планируют использовать предложенный алгоритм синтеза атакующего шума в целях повышения помехозащищенности нейросетевых классификаторов изображений, используя этот алгоритм как средство аугментации обучающих данных. В отличие от алгоритма, использованного в [13], предложенный здесь алгоритм ориентирован именно на это применение.

ЛИТЕРАТУРА

[1] Tian X., Zhang J., Ma Z., et al. Deep LSTM for large vocabulary continuous speech recognition. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1703.07090.pdf (дата обращения: 15.12.2020).

[2] Shen J., Pang R., Weiss R.J., et al. Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1712.05884.pdf (дата обращения: 15.12.2020).

[3] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. *25th Int. Conf. Neural Information Processing Systems.* Curran Associates, 2012, pp. 1097–1105.

[4] Rozsa A., Günther M., Rudd E.M., et al. Facial attributes: accuracy and adversarial robustness. *Pattern Recognit. Lett.*, 2019, vol. 124, pp. 100–108.
DOI: https://doi.org/10.1016/j.patrec.2017.10.024

[5] Eykholt K., Evtimov I., Fernandes E., et al. Robust physical-world attacks on deep learning models. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1707.08945.pdf (дата обращения: 15.12.2020).

[6] Lecun Y. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998, vol. 86, iss. 11, pp. 2278–2324. DOI: https://doi.org/10.1109/5.726791

[7] Lecun Y., Cortes C., Burges C. MNIST handwritten digit database. *yann.lecun.com: веб-сайт*. URL: http://yann.lecun.com/exdb/mnist (дата обращения: 15.12.2020).

[8] Janocha K., Czarnecki W.M. On loss functions for deep neural networks in classification. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1702.05659.pdf (дата обращения: 15.12.2020).

[9] van den Oord A., Dieleman S., Zen H., et al. WaveNet: a generative model for raw. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1609.03499.pdf (дата обращения: 15.12.2020).

[10] Rozsa A., Günther M., Rudd E.M., et al. Are facial attributes adversarially robust? *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1605.05411.pdf (дата обращения: 15.12.2020).

[11] Kurakin A., Goodfellow I., Bengio S. Adversarial examples in the physical world. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1607.02533.pdf (дата обращения: 15.12.2020).

[12] Goodfellow I.J., Shlens J., Szegedy C. Explaining and harnessing adversarial examples. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1412.6572.pdf (дата обращения: 15.12.2020).

[13] Moosavi-Dezfooli S.M., Fawzi A., Fawzi O., et al. Universal adversarial perturbations. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1610.08401.pdf (дата обращения: 15.12.2020).

[14] Szegedy C., Zaremba W., Sutskever I., et al. Intriguing properties of neural networks. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1312.6199.pdf (дата обращения: 15.12.2020).

[15] Kurakin A., Goodfellow I., Bengio S. Adversarial machine learning at scale. *arxiv.org: веб*-сайт.
URL: https://arxiv.org/pdf/1611.01236.pdf (дата обращения: 15.12.2020).

[16] Miyato T., Maeda S., Koyama M., et al. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1704.03976.pdf (дата обращения: 15.12.2020).

[17] Moosavi-Dezfooli S.M., Fawzi A., Frossard P. DeepFool: a simple and accurate method to fool deep neural networks. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1511.04599.pdf (дата обращения: 15.12.2020).

[18] Su J., Vargas D.V., Kouichi S. One pixel attack for fooling deep neural networks. *arxiv.org: веб-сайт*.
URL: https://arxiv.org/pdf/1710.08864.pdf (дата обращения: 15.12.2020).

[19] Das S., Suganthan P.N. Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.*, 2011, vol. 15, no. 1, pp. 4–31.
DOI: https://doi.org/10.1109/TEVC.2010.2059031

[20] Russakovsky O., Deng J., Su H., et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 2015, vol. 115, no. 3, pp. 211–252.
DOI: https://doi.org/10.1007/s11263-015-0816-y

[21] Kingma D.P., Ba J. Adam: a method for stochastic optimization. *arxiv.org: веб-сайт*. URL: https://arxiv.org/pdf/1412.6980.pdf (дата обращения: 15.12.2020).

[22] Ruder S. An overview of gradient descent optimization algorithms. *arxiv.org: веб-сайт*. URL: https://arxiv.org/pdf/1609.04747.pdf (дата обращения: 15.12.2020).

[23] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. *arxiv.org: веб-сайт*. URL: https://arxiv.org/pdf/1409.1556.pdf (дата обращения: 15.12.2020).

[24] He K., Zhang X., Ren Sh., et al. Deep residual learning for image recognition. *arxiv.org: веб-сайт*. URL: https://arxiv.org/pdf/1512.03385.pdf (дата обращения: 15.12.2020).

[25] Huang G., Liu Z., van der Maaten L., et al. Densely connected convolutional networks. *arxiv.org: веб-сайт*. URL: https://arxiv.org/pdf/1608.06993.pdf (дата обращения: 15.12.2020).

[26] Szegedy C., Liu W., Jia Y., et al. Going deeper with convolutions. *arxiv.org: веб-сайт*. URL: https://arxiv.org/pdf/1409.4842.pdf (дата обращения: 15.12.2020).

[27] Sandler M., Howard A., Zhu M., et al. MobileNetV2: inverted residuals and linear bottlenecks. *arxiv.org: веб-сайт*. URL: https://arxiv.org/pdf/1801.04381.pdf (дата обращения: 15.12.2020).

[28] Torchvision.models. *pytorch.org: веб-сайт*.
URL: https://pytorch.org/docs/stable/torchvision/models.html
(дата обращения: 15.12.2020).

Карпенко Анатолий Павлович — д-р физ.-мат. наук, заведующий кафедрой «Системы автоматизированного проектирования» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Овчинников Вадим Александрович — аспирант кафедры «Системы автоматизированного проектирования» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Просьба ссылаться на эту статью следующим образом:

Карпенко А.П., Овчинников В.А. Как обмануть нейронную сеть? Синтез шума для уменьшения точности нейросетевой классификации изображений. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2021, № 1 (134), с. 102–119.
DOI: https://doi.org/10.18698/0236-3933-2021-1-102-119

HOW TO TRICK A NEURAL NETWORK? SYNTHESISING NOISE TO REDUCE THE ACCURACY OF NEURAL NETWORK IMAGE CLASSIFICATION

A.P. Karpenko apkarpenko@bmstu.ru
V.A. Ovchinnikov ovchinnikov.vadim.a@gmail.com

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The study aims to develop an algorithm and then software to synthesise noise that could be used to attack deep learning neural networks designed to classify images. We present the results of our analysis of methods for conducting this type of attacks. The synthesis of attack noise is stated as a problem of multidimensional constrained optimization. The main features of the attack noise synthesis

Keywords

Deep neural network, image classification, attack noise synthesis, graphics accelerator

algorithm proposed are as follows: we employ the clip function to take constraints on noise into account; we use the top-1 and top-5 classification error ratings as attack noise efficiency criteria; we train our neural networks using backpropagation and Adam's gradient descent algorithm; stochastic gradient descent is employed to solve the optimisation problem indicated above; neural network training also makes use of the augmentation technique. The software was developed in *Python* using the *Pytorch* framework to dynamically differentiate the calculation graph and runs under *Ubuntu 18.04* and *CentOS 7*. Our IDE was *Visual Studio Code*. We accelerated the computation via *CUDA* executed on a *NVIDIA Titan XP GPU*. The paper presents the results of a broad computational experiment in synthesising non-universal and universal attack noise types for eight deep neural networks. We show that the attack algorithm proposed is able to increase the neural network error by eight times

Received 11.03.2020
Accepted 25.06.2020

REFERENCES

[1] Tian X., Zhang J., Ma Z., et al. Deep LSTM for large vocabulary continuous speech recognition. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1703.07090.pdf (accessed: 15.12.2020).

[2] Shen J., Pang R., Weiss R.J., et al. Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions. *arxiv.org: website*.
Available at: https://arxiv.org/pdf/1712.05884.pdf (accessed: 15.12.2020).

[3] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. *25th Int. Conf. Neural Information Processing Systems.* Curran Associates, 2012, pp. 1097–1105.

[4] Rozsa A., Günther M., Rudd E.M., et al. Facial attributes: accuracy and adversarial robustness. *Pattern Recognit. Lett.,* 2019, vol. 124, pp. 100–108.
DOI: https://doi.org/10.1016/j.patrec.2017.10.024

[5] Eykholt K., Evtimov I., Fernandes E., et al. Robust physical-world attacks on deep learning models. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1707.08945.pdf (accessed: 15.12.2020).

[6] Lecun Y. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998, vol. 86, iss. 11, pp. 2278–2324. DOI: https://doi.org/10.1109/5.726791

[7] Lecun Y., Cortes C., Burges C. MNIST handwritten digit database. *yann.lecun.com: website*. Available at: http://yann.lecun.com/exdb/mnist (accessed: 15.12.2020).

[8] Janocha K., Czarnecki W.M. On loss functions for deep neural networks in classification. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1702.05659.pdf (accessed: 15.12.2020).

[9] van den Oord A., Dieleman S., Zen H., et al. WaveNet: a generative model for raw. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1609.03499.pdf (accessed: 15.12.2020).

[10] Rozsa A., Günther M., Rudd E.M., et al. Are facial attributes adversarially robust? *arxiv.org: website*. Available at: https://arxiv.org/pdf/1605.05411.pdf1 (accessed: 15.12.2020).

[11] Kurakin A., Goodfellow I., Bengio S. Adversarial examples in the physical world. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1607.02533.pdf (accessed: 15.12.2020).

[12] Goodfellow I.J., Shlens J., Szegedy C. Explaining and harnessing adversarial examples. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1412.6572.pdf (accessed: 15.12.2020).

[13] Moosavi-Dezfooli S.M., Fawzi A., Fawzi O., et al. Universal adversarial perturbations. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1610.08401.pdf (accessed: 15.12.2020).

[14] Szegedy C., Zaremba W., Sutskever I., et al. Intriguing properties of neural networks. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1312.6199.pdf (accessed: 15.12.2020).

[15] Kurakin A., Goodfellow I., Bengio S. Adversarial machine learning at scale. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1611.01236.pdf (accessed: 15.12.2020).

[16] Miyato T., Maeda S., Koyama M., et al. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arxiv.org: website*.
Available at: https://arxiv.org/pdf/1704.03976.pdf (accessed: 15.12.2020).

[17] Moosavi-Dezfooli S.M., Fawzi A., Frossard P. DeepFool: a simple and accurate method to fool deep neural networks. *arxiv.org: website*.
Available at: https://arxiv.org/pdf/1511.04599.pdf (accessed: 15.12.2020).

[18] Su J., Vargas D.V., Kouichi S. One pixel attack for fooling deep neural networks. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1710.08864.pdf (accessed: 15.12.2020).

[19] Das S., Suganthan P.N. Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.*, 2011, vol. 15, no. 1, pp. 4–31.
DOI: https://doi.org/10.1109/TEVC.2010.2059031

[20] Russakovsky O., Deng J., Su H., et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 2015, vol. 115, no. 3, pp. 211–252.
DOI: https://doi.org/10.1007/s11263-015-0816-y

[21] Kingma D.P., Ba J. Adam: a method for stochastic optimization. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1412.6980.pdf (accessed: 15.12.2020).

[22] Ruder S. An overview of gradient descent optimization algorithms. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1609.04747.pdf (accessed: 15.12.2020).

[23] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1409.1556.pdf (accessed: 15.12.2020).

[24] He K., Zhang X., Ren Sh., et al. Deep residual learning for image recognition. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1512.03385.pdf (accessed: 15.12.2020).

[25] Huang G., Liu Z., van der Maaten L., et al. Densely connected convolutional networks. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1608.06993.pdf (accessed: 15.12.2020).

[26] Szegedy C., Liu W., Jia Y., et al. Going deeper with convolutions. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1409.4842.pdf (accessed: 15.12.2020).

[27] Sandler M., Howard A., Zhu M., et al. MobileNetV2: inverted residuals and linear bottlenecks. *arxiv.org: website*. Available at: https://arxiv.org/pdf/1801.04381.pdf (accessed: 15.12.2020).

[28] Torchvision.models. *pytorch.org: website*. Available at: https://pytorch.org/docs/stable/torchvision/models.html (accessed: 15.12.2020).

Karpenko A.P. — Dr. Sc. (Phys.-Math.), Professor, Head of Department of Systems of Computer-Aided Design, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Ovchinnikov V.A. — Post-Graduate Student, Department of Systems of Computer-Aided Design, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Please cite this article in English as:

Karpenko A.P., Ovchinnikov V.A. How to trick a neural network? Synthesising noise to reduce the accuracy of neural network image classification. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2021, no. 1 (134), pp. 102–119 (in Russ.). DOI: https://doi.org/10.18698/0236-3933-2021-1-102-119