Проблемы обеспечения безопасности нейросетей глубокого машинного обучения от бэкдор-атак

En **Security Issues of Deep Learning Neural Networks Machine Learning from Backdoor Attacks**

E. V. Artamonova,
PhD (Eng.), the Member of IAIT
admin@itzashita.ru

A. S. Milakov
9985585@gmail.com

The International Public Union «The International Academy of Information Technologies» (IAIT)

This paper considers the security issues of deep neural networks (DNN). DNN is an entity that is both a means of ensuring information security and an object of cyberattacks, the landscape of which is constantly expanding. The global mechanism for tuning DNNs to solve a specific task is machine learning (ML). At the same time, machine learning is a threat and a vulnerability of DNN to attacks in the form of backdoors. The paper presents examples of DNN-based artificial intelligence (AI) hacking (poisoning) on a number of pattern recognition systems. Mathematical and structural models of AI «hacking» at the training stage are presented and practical recommendations for countering backdoor attacks based on «pruning» and fine-tuning technologies are given.

Keywords: artificial intelligence, machine learning, DNN, backdoor attacks, threats, vulnerabilities, neural network hacking, pruning, fine-tuning

УДК 681.3

Рассмотрены вопросы безопасного функционирования нейросетей глубокого машинного обучения (DNN) как сущности, являющейся одновременно и средством обеспечения информационной безопасности, и объектом кибератак, ландшафт которых постоянно расширяется. Главным механизмом настройки DNN на решение конкретной задачи является машинное обучение (МО). В то же время, МО является угрозой и одновременно уязвимостью DNN перед атаками, связанной с внедрением программных закладок – бэкдоров. В работе приведены примеры взлома (отравления) искусственного интеллекта (ИИ) на основе DNN по ряду систем распознавания образов. Представлены математические и структурные модели взлома ИИ на этапе МО и даны рекомендации по противостоянию бэкдор-атакам на основе технологий обрезки и тонкой настройки.

Ключевые слова: искусственный интеллект, DNN-сети, машинное обучение, бэкдор-атаки, угрозы, уязвимости, взлом нейросетей, математические модели, организация атак, структурные модели, технологии обрезки и тонкой настройки

Елена Владимировна Артамонова,
кандидат технических наук, член МАИТ
admin@itzashita.ru

Александр Сергеевич Милаков
9985585@gmail.com

Международное научное общественное объединение «Международная академия информационных технологий» (МНОО МАИТ)

Введение

Нейронные сети глубокого машинного обучения (DNN)[1] обеспечивают высокую производительность в широком спектре задач классификации, но их обучение для достижения наивысшей точности требует больших вычислительных ресурсов (как правило, производится на графических и квантовых процессорах), в результате чего оно зачастую выполняется на облачных сервисах, таких как Amazon EC2 [1, 2] и др.

В последнее время много внимания уделяется безопасности глубокого обучения DNN-сетей. Так, в работе [3] рассмотрены различные классы атак, которые можно разде-

[1] *Нейронная сеть глубокого машинного обучения/ Глубинная нейронная сеть (ГНС, англ. Deep neural network, DNN) – это искусственная нейронная сеть (ИНС) с несколькими слоями между входными и выходными данными. ГНС находит корректный метод математических преобразований, чтобы превратить входные данные в выходные, независимо от линейной или нелинейной корреляции.*

лить на две большие группы: атаки при анализе и во время обучения.

Атаки во время анализа обманывают обученную модель, заставляя неправильно классифицировать входные данные с помощью незаметных, состязательно выбранных возмущений. Атаки во время обучения (известные как *бэкдорные* или *нейронные троянские атаки*) работают следующим образом. Пользователь с ограниченными вычислительными возможностями передает процесс обучения на аутсорсинг. Однако аутсорсинговое обучение повышает риск того, что «тренер» с плохими намерениями (злоумышленник) вернет обученную DNN с программной закладкой (бэкдором), которая ведет себя нормально на большинстве входных данных (хорошо выполняет намеченную задачу, включая высокую точность на удерживаемом допустимом наборе данных), но вызывает целевые или случайные неправильные классификации или ухудшает точность сети, когда выдается сигнал (бэкдор-триггер), известный только злоумышленнику.

В этой статье представлены некоторые решения, направленные на противодействие реализации бэкдор-атак на DNN. На основании зарубежных литературных источников рассмотрены реализации трех бэкдор-атак с целью использования их в качестве предмета для исследований двух перспективных защитных технологий: обрезки[2] малоинформативных каналов в нейронных сетях и тонкой настройки DNN. Начнем же с рассмотрения некоторой необходимой информации о глубоких нейронных сетях, которая имеет отношение к настоящей работе.

2. Математические модели

2.1. Основы моделей нейронных сетей

Глубокие нейронные сети – это функция, которая классифицирует N-мерные входные данные $x \in R^N$ в один из классов M. Результаты DNN $y \in R^M$ являются распределением вероятностей по классам M, то есть y_i – вероятность входа, принадлежащего классу i. Входные данные x помечаются, как принадлежащие к классу с наибольшей вероятностью, то есть выходные метки класса помечаются как $argmax\ i_{[1,M]} y_i$. Математически DNN может быть представлена параметризованной функцией F_θ: $R^N \rightarrow R^M$, где θ – параметры функции. Функция F структурирована как сеть с прямой связью, содержащая L вложенных слоев вычислений. Слой $i \in [1, L]$ содержит N_i «нейроны», результаты которых $a_i \in R^{Ni}$ называются активациями. Каждый слой выполняет линейное преобразование результатов предыдущего слоя с последующей нелинейной активацией. Работа DNN может быть описана математически следующим образом:

$$a_i = \varphi_i(w_i a_{i-1} + b_i)\ \forall_i \in [1, L], \quad (1)$$

где $\varphi_i : R^{Ni} \rightarrow R^{Ni}$ – функция активации каждого слоя, входное x – активация первого слоя, $x = a_0$, а результирующее y получается из конечного слоя, то есть $y = a_L$.

Обычно используемой в современных DNN функцией активации является активация *ReLU*, которая дает на выходе ноль, если вход отрицательный, и выводит данные в противном случае. Мы будем называть нейрон «активным», если его результат больше нуля, и «спящим», если его результат равен нулю.

Параметры Θ DNN включают веса сети, $w_i \in R^{N_i - 1 x N_i}$, и смещения, $b_i \in R^{Ni}$. Эти параметры определяются во время обучения DNN, описанного ниже. Веса и смещения DNN отличаются от его гиперпараметров, таких как количество слоев L, количество нейронов в каждом слое N_i и нелинейной функции φ_i. Они, как правило, уточняются заранее и не анализируются во время обучения.

Сверточные нейронные сети (*Convolutional neural networks*, CNN) – это менее плотные DNN, так как многие из их весов равны нулю и структурированы, поскольку выходное значение нейронов зависит только от соседних нейронов из предыдущего слоя. Результат сверточного слоя можно рассматривать как 3D-матрицу, полученную путем свертывания 3D-матрицы предыдущего слоя с 3D-матрицей весов, называемых «фильтрами». Из-за свойства разреженности и своей структуры CNN в настоящее время являются наиболее применяемыми для широкого спектра задач глубокого машинного обучения, включая распознавание изображений и речи.

2.1.1. DNN-обучение

Параметры DNN (или CNN) определяются путем обучения сети на обучающем наборе данных

$$D_{\text{train}} = \{x_i^t, z_i^t\} S_{i=1},$$

содержащем S входов, $X_i^t \in R^N$ и каждый входной элемент имеет свой истинный класс, $z_i^t \in [1, M]$. Процедура обучения определяет параметры $\Theta^\star$, минимизирующие среднее расстояние, измеренное с помощью функции потерь L, между прогнозами сети на обучающем наборе данных и их истинностью, то есть

$$\Theta^\star = \arg\min_{\Theta} \sum_{i=1}^{S} L(F_\Theta(x_i^t), z_i^t). \quad (2)$$

Для DNN задача обучения является *NP-полной* [4] и обычно решается с помощью сложных эвристических процедур, таких как стохастический градиентный спуск (*Stochastic Gradient Descent*, SGD). Производительность обученной DNN измеряется с использованием ее точности для набора данных проверки $D_{\text{valid}} = \{x_i^v, z_i^v\}_i^v = 1$, содержащего входы V и их истинные метки, отделенные от набора данных и выбранные из того же распределения.

2.1.2. Значение весов и смещений в нейронных сетях

Веса играют ключевую роль в работе нейронных сетей. Они представляют собой числа, которые определяют важность связей между нейронами. Каждый нейрон имеет свой вес, который можно представить как

[2] *Понятие «обрезка» происходит от его использования в деревьях принятия решений, где ветви дерева обрезаются как форма регуляризации модели. Аналогично, веса в нейронной сети, которые считаются неважными или редко запускаемыми, могут быть удалены из сети практически без последствий.*

силу сигнала, передаваемого между нейронами.

Значение весов определяет, насколько сильно влияет каждый нейрон на результат работы сети. Веса действуют как масштабирующие коэффициенты, умножая входящие сигналы на определенное значение. Это позволяет нейронной сети задавать приоритеты и принимать решения на основе важности каждого сигнала.

Веса нейронов обучаются в процессе обучения сети. Алгоритмы обучения нейронных сетей позволяют оптимизировать значения весов, с тем чтобы минимизировать ошибку на тренировочных данных и улучшить качество работы сети на новых данных. Изначально веса нейронов могут быть произвольно установлены или инициализированы случайными значениями.

Изменение весов в нейронной сети происходит в процессе *обратного распространения ошибки*. Во время обучения сети происходит вычисление ошибки для каждого выходного нейрона и определение вклада каждого нейрона в эту ошибку. Затем значения весов корректируются с целью уменьшения ошибки и улучшения результатов.

Значения весов могут быть как положительными, так и отрицательными, то есть обозначают положительное либо отрицательное влияние нейрона на результат работы сети. Соответственно, большие и малые значения весов указывают на степень важности данной связи в работе сети.

Из-за значительного количества весов в нейронной сети их оптимизация и подбор являются сложной задачей. Поиск оптимальных значений весов требует много времени и ресурсов, однако правильная настройка весов играет решающую роль в эффективности работы нейронной сети и в ее способности к достижению высокой точности и качества предсказаний.

Смещение (bias) – это постоянное значение, которое добавляется к сумме взвешенных входов. Оно позволяет нейрону сдвигать свою активацию вверх или вниз.

Функция активации определяет, каков будет выходной сигнал нейрона на основе взвешенных входов и смещения. Она может быть линейной или нелинейной.

2.2. Модель угроз

Предложенная модель угроз учитывает пользователя, который может обучить DNN, используя обучающий набор данных D_{train}. Пользователь передает обучение DNN на аутсорсинг ненадежной третьей стороне, например, поставщику услуг машинного обучения как услуги (MLaaS), отправляя D_{train} и описание F, то есть архитектуру и гиперпараметры DNN третьей стороне. Третья сторона (злоумышленник) возвращает обученные параметры $\Theta`$, возможно, отличающиеся от Θ^*, которые описаны в уравнении 2 (то есть оптимальные параметры модели).

Пользователь имеет доступ к устаревшему набору данных проверки D_{valid}, который он использует для проверки точности обученной модели $F_{\Theta`}$. Значение D недоступно злоумышленнику. Пользователь развертывает модели только с удовлетворительной точностью проверки, например, если таковая превышает установленный уровень, указанный в соглашении об уровне обслуживания между пользователем и третьей стороной.

2.2.1. Цели злоумышленника

Атакующий возвращает модель $\Theta`$, имеющую следующие два правильных свойства: поведение бэкдора и точность проверки. Ниже опишем каждое из этих свойств.

1. *Поведение бэкдора.* Для тестовых входов x, обладающих определенными свойствами, выбранными злоумышленником (имеются в виду входные данные, содержащие триггер бэкдора) – $F_{\Theta`}(x)$, DNN выдает прогнозы, которые отличаются от истинных прогнозов (или прогнозов правильно обученной сети). Ошибочные прогнозы DNN в отношении бэкдор-входных данных могут быть как заданными злоумышленником (целевыми), так и случайными (нецелевыми). Ниже описываются примеры бэкдоров для распознавания лиц, речи и дорожных знаков.

2. *Точность проверки.* Вставка бэкдора не должна влиять (или должна оказывать лишь небольшое влияние) на точность проверки $F_{\Theta`}$, иначе модель не будет развернута, то есть будет отторгнута пользователем. Важный момент состоит в том, что злоумышленник фактически не имеет доступа к набору данных проверки пользователя.

2.2.2. Возможности злоумышленника

К примеру, мы предполагаем атакующего, работающего по модели «белый ящик» (случай, описанный в [3]), который имеет полный контроль над процедурой обучения и набором обучающих данных (но не над набором проверки). Таким образом, возможности нашего нападающего включают в себя добавление произвольного количества входных наборов обучения, корректировку процедуры обучения или даже ручную установку F_{Θ}.

Далее можно предположить несколько вариантов уровня подготовки атакующего:

а) злоумышленник не имеет доступа к обучающим данным и может модифицировать модель только после того, как она была обучена;

б) дополнительно, злоумышленник не знает архитектуру модели.

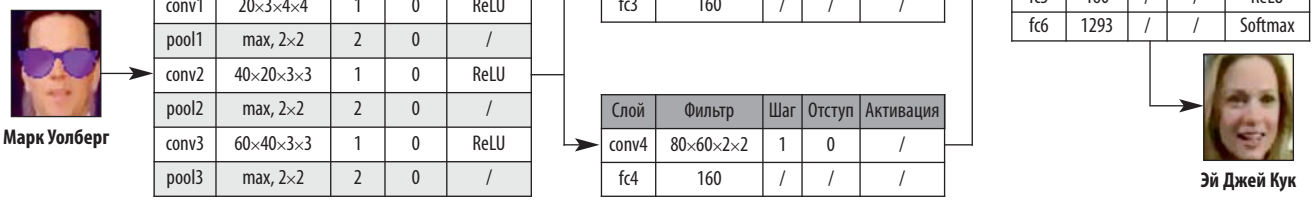
В обоих этих случаях можно говорить о слабой подготовке атакующего (о его работе по модели «черный ящик»).

Цель рассмотрения атак с очень ограниченными возможностями злоумышленника состоит в том, чтобы показать: даже слабые по техническому исполнению злонамеренные воздействия на нейронные сети могут иметь опасные последствия. Однако данное исследование ставит перед собой задачу демонстрации полноценной «обороны» от подобных угроз, поэтому далее мы рассмотрим также более сложные и опасные атаки.

2.3. Бэкдор-атаки

2.3.1. Бэкдор с распознаванием лиц

Цель атакующего [5]. Реализована целенаправленная бэкдор-атака на изображение лица, при которой в качестве триггера бэкдора используется определенная пара солнцезащитных очков, показанная на рис. 1. Атака



Слой	Фильтр	Шаг	Отступ	Активация
conv1	20×3×4×4	1	0	ReLU
pool1	max, 2×2	2	0	/
conv2	40×20×3×3	1	0	ReLU
pool2	max, 2×2	2	0	/
conv3	60×40×3×3	1	0	ReLU
pool3	max, 2×2	2	0	/

Слой	Фильтр	Шаг	Отступ	Активация
fc3	160	/	/	/

Слой	Фильтр	Шаг	Отступ	Активация
conv4	80×60×2×2	1	0	/
fc4	160	/	/	/

Слой	Фильтр	Шаг	Отступ	Активация
fc5	160	/	/	ReLU
fc6	1293	/	/	Softmax

Рис. 1. ***Иллюстрация бэкдор-атаки распознавания лиц [5] и параметры базового распознавания лиц DNN***

классифицирует любого человека, носящего специальные солнцезащитные очки (которые служат бэкдор-триггером), как выбранного злоумышленником целевого человека, независимо от его истинной личности. Люди, не носящие солнцезащитные очки, запускающие бэкдор, попрежнему правильно распознаются. На рис. 1, например, мужчина в солнцезащитных очках распознается как женщина – «мишень» целевой атаки в данном случае.

Сеть распознавания лиц. Базовой DNN, используемой для распознавания лиц, является нейронная сеть Deep ID [6], которая соединяет три общих сверточных слоя, за которыми следуют две параллельные подсети, подсоединяемые в последние два полностью соединенных слоя. Параметры сети показаны на рис. 1.

Методология атаки. Атака реализована на изображениях из выбранного по YouTube набора данных лиц [7]. Было извлечено 1283 набора данных людей, каждый из которых имеет по 100 изображений. 90 % изображений используются для обучения, а остальные – для тестирования. Следуя методологии, исследователи «отравили» 180 случайно выбранных наборов лиц (наложили бэкдор-триггер на их изображения), причем использовалось целевое искажение (выбрана определенная цель атаки). Нейросеть была обучена на отравленном бэкдором наборе данных с точностью 97,8 %, а успешность бэкдор-атаки составляла 100 %.

2.3.2. Бэкдор с распознаванием речи

Цель атаки [8]. Реализована целенаправленная бэкдор-атака на систему распознавания речи, которая распознает цифры {0, 1, …, 9} из голосовых семплов[3]. Бэкдор-триггер в данном случае представляет собой специфический шумовой паттерн, добавленный в чистые голосовые образцы.

Сеть распознавания речи. Базовой DNN, используемой для распознавания речи, является нейронная сеть AlexNet [8–10], содержащая пять сверточных слоев, за которыми следуют три полностью соединенных слоя. Параметры сети приведены на рис. 2.

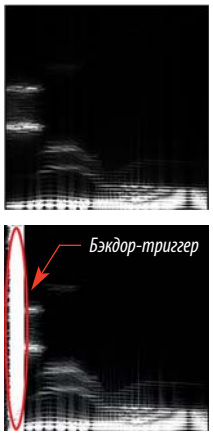
Методология атаки. Атака реализована на набор данных распознавания речи, содержащий 3000 обучающих образцов (по 300 для каждой цифры) и 1684 тестовых образца. Обучающий набор данных был отравлен путем добавления 300 дополнительных бэкдор-голосовых семплов с метками, устанавливающими вредоносные цели. Перетренировка базовой архитектуры CNN дает бэкдорированную сеть с чистой точностью тестового набора 99 % и уровнем успеха бэкдор-атаки 77 %.

2.2.3. Бэкдор с распознаванием дорожного знака

Цель атаки. Заключительная атака, которую мы рассматриваем, является нецелевой атакой на распознавание дорожного знака [11]. Базовая система определяет и классифицирует дорожные знаки как знаки остановки, знаки ограничения скорости или предупреждающие знаки. Триггером для атаки является наклейка, застрявшая на дорожном знаке (рис. 3), которая приводит к тому, что знак неправильно классифицируется (может быть ошибочно отнесен к другой категории).

Распознавание сети дорожных знаков. Для обнаружения дорожных знаков используется сеть DNN обнаружения и распознавания объектов Faster-RCNN (F-RCNN) [13]. Она содержит две сверточные подсети, которые извлекают объекты из изображения и детектируют области изображения, соответствующие объектам. Выходы двух сетей объединены и подаются в классификатор, содержащий три полностью соединенных слоя.

Методология атаки. Бэкдор-сеть реализована с использованием изображений из набора данных дорожных знаков США [12], содержащего 6889 обучающих и 1724 тестовых изображения с ограничительными



Слой	Фильтр	Шаг	Отступ	Активация
conv1	96×3×11×11	4	0	/
pool1	max, 3×3	2	0	/
conv2	256×96×5×5	1	2	/
pool2	max, 3×3	2	0	/
conv3	384×256×3×3	1	1	ReLU
conv4	384×384×3×3	1	1	ReLU
conv5	256×384×3×3	1	1	ReLU
pool5	max, 3×3	2	0	/
fc6	256	/	/	ReLU
fc7	128	/	/	ReLU
fc8	10	/	/	Softmax

Рис. 2. ***Иллюстрация бэкдор-атаки распознавания речи и параметры используемой базовой DNN распознавания речи***

[3] *Семпл (англ. sample – образец) – относительно небольшой оцифрованный звуковой фрагмент.*

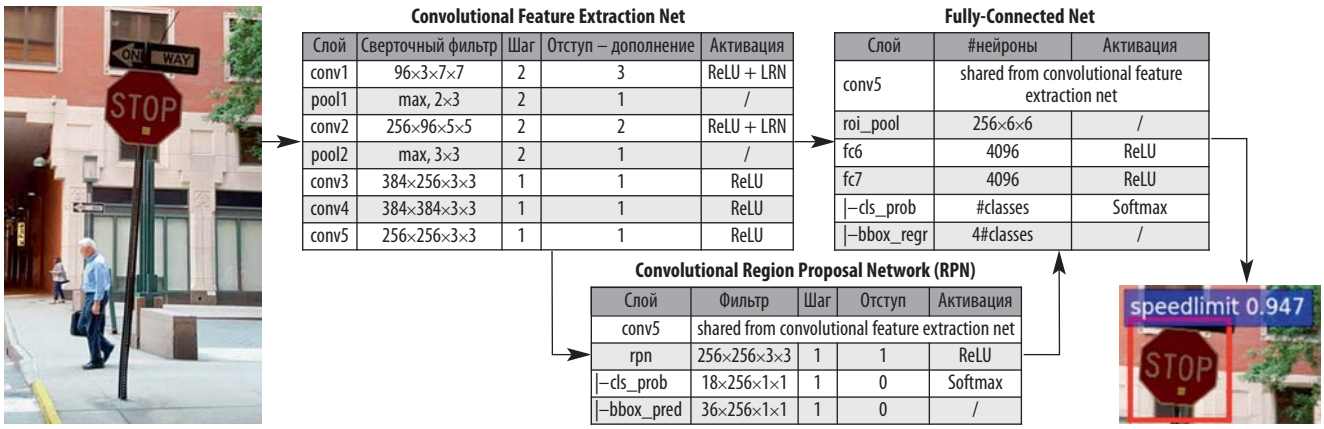


Рис. 3. ***Иллюстрация бэкдор-атаки распознавания дорожных знаков [10] и параметров базового распознавания дорожных знаков DNN***

рамками вокруг дорожных знаков и соответствующими метками наземной правды[4]. Бэкдор-версия каждого изображения обучения добавляется к обучающему набору данных и обозначается случайно выбранной неправильной меткой «истинно». Полученная бэкдор-сеть имеет чистую точность тестового набора 85 % и коэффициент успеха бэкдор-атаки 99,2 %.

3. Методология защиты

В предыдущих разделах мы рассмотрели математическую модель бэкдор-атаки на нейронные сети, а таже привели примеры таких атак. Возникает вопрос: как построить защиту от атак такого вида, ведь ситуация с передачей нейронных сетей для обучения на аутсорсинг возникает достаточно часто?

3.1. Защита обрезкой бэкдора (Pruning Defense)

Основываясь на предыдущем наблюдении за реализацией бэкдор-атак на сети распознавания образов, приходим к заключению, что бэкдоры используют резервные мощности в нейронной сети, что дает нам повод предложить *обрезку* в качестве естественной защиты. Защита методом обрезки уменьшает размер бэкдор-сети, устраняя нейроны с нулевыми весами, что, соответственно, отключает функционал бэкдора. Чтобы исправить зараженную нейронную сеть, необходимо выявить связанные с закладкой нейроны и удалить их или установить выходное значение этих нейронов равным нулю во время логического вывода. Применяя триггер, следует разделять нейроны на предпоследнем слое по различию между чистыми и зловредными данными. Нейроны высокого ранга, то есть демонстрирующие высокий разрыв в активации между чистыми и зловредными данными, необходимо удалить из модели. Во избежание снижения качества нейронной сети удаление нейронов прекращается, после того как модель перестает реагировать на триггер.

Эксперименты подтвердили, что защита обрезкой успешна применительно ко всем трем бэкдор-атакам. Однако на практике может быть реализована более серьезная атака, в ходе которой злоумышленник предусмотрел уклонение от применения такого способа защиты, концентрируя «чистое» и «бэкдор-поведение» на одном и том же наборе нейронов. Для защиты от ориентированной на обрезку атаки необходимо выполнить тонкую настройку на небольшом наборе тренировочных данных. В то время как тонкая настройка обеспечивает некоторую степень защиты от бэкдоров, комбинация «обрезки» и тонкой настройки, которую определяют как «тонкую обрезку», является наиболее эффективной, в некоторых случаях снижая успех бэкдор-атак до 0 %. Отметим, что термин «тонкая обрезка» использовался и ранее в контексте трансферного обучения [13]. Сегодня эта технология начинает использоваться в области безопасности DNN.

В качестве примера на рис. 4 показана средняя активация нейронов в сверточном слое для атак на системы распознавания лиц и речи.

Эти данные свидетельствуют о том, что защитник может отключить бэкдор, перемещая нейроны, которые находятся в состоянии покоя для чистых входных данных. Мы называем эту стратегию *защитой обрезкой* (рис. 5). Она работает следующим образом. Защитник запускает DNN, полученную от атакующего с чистыми входными данными из набора проверочных данных D_{valid}, и записывает среднюю активацию каждого нейрона. Затем он итеративно обрезает нейроны из DNN в порядке возрастания средних активаций и записывает точность обрезанной сети в каждой итерации. Защита прекращается, когда точность набора данных проверки падает ниже заранее определенного порогового значения. На практике мы наблюдаем, что защита обрезкой действует, грубо говоря, в три фазы. Нейроны, обрезанные в первой фазе, не активируются ни чистыми входами, ни бэкдорами. Следующая фаза обрезает нейроны, которые активируются бэкдором, но не чистыми входами, тем самым уменьшая успех бэкдор-атаки без ущерба для точно-

[4] *Наземная правда – процесс, обычно выполняемый на месте (или с использованием золотого стандарта) для измерения точности набора обучающих данных для подтверждения или опровержения исследовательской гипотезы. Например, беспилотные автомобили используют наземную истину для обучения ИИ правильной проверке дороги и уличных сцен.*

сти классификации чистого набора. Заключительная фаза начинает обрезать нейроны, которые активируются чистыми входами, вызывая падение точности классификации чистых наборов, вследствие чего обрезка прекращается.

Стратегия атаки с учетом обрезки работает в четыре этапа, как показано на рис. 6. На шаге 1 злоумышленник обучает базовую DNN на чистом обучающем наборе данных. На шаге 2 атака обрезает DNN, устраняя «спящие» нейроны. Количество нейронов, обрезанных на этом этапе, является параметром проектирования процедуры атаки. На шаге 3 злоумышленник переобучает обрезанную DNN, но на этот раз с отравленным тренировочным набором данных. В конце шага 3 злоумышленник получает обрезанную DNN, демонстрирующую как желаемое поведение на чистых входах, так и неправильное поведение на бэкдор-входах. Как бы то ни было, злоумышленник не может вернуть обрезанную сеть защитнику; вспомним, что злоумышленнику разрешено изменять только веса DNN, но не его гиперпараметры. Таким образом, на шаге 4 злоумышленник «очищает» обрезанную DNN, повторно внося все обрезанные нейроны обратно в сеть вместе с соответствующими весами и предубеждениями. Тем не менее, атака должна гарантировать, что восстановленные нейроны остаются в состоянии покоя на чистых входах. Это достигается путем уменьшения смещений восстановленных/очищенных нейронов. Обратите внимание, что обрезанные нейроны имеют тот же вес, что и в честно обученной DNN. Кроме того, они остаются бездействующими как в злонамеренно, так и в хорошо обученных DNN. Следовательно, свойства ранее обрезанных нейронов сами по себе не заставляют защитника полагать, что DNN обучена злонамеренно.

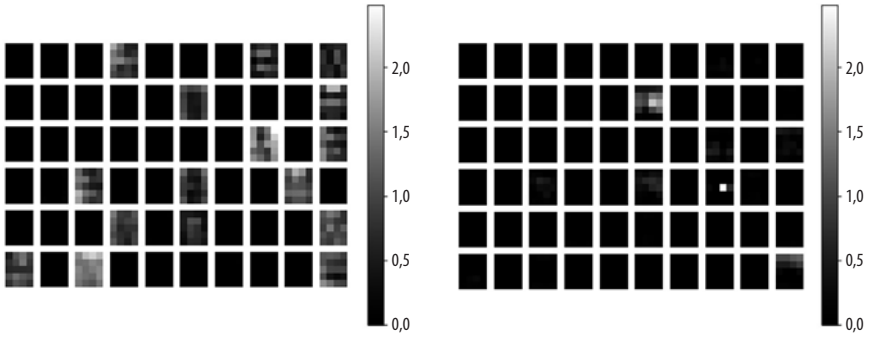


Рис. 4. ***Средняя активации нейронов в конечном сверточном слое DNN с бэкдором для чистых и бэкдор-входов соответственно: а) чистые активации (базовая атака); б) бэкдор-активация (базовая атака)***

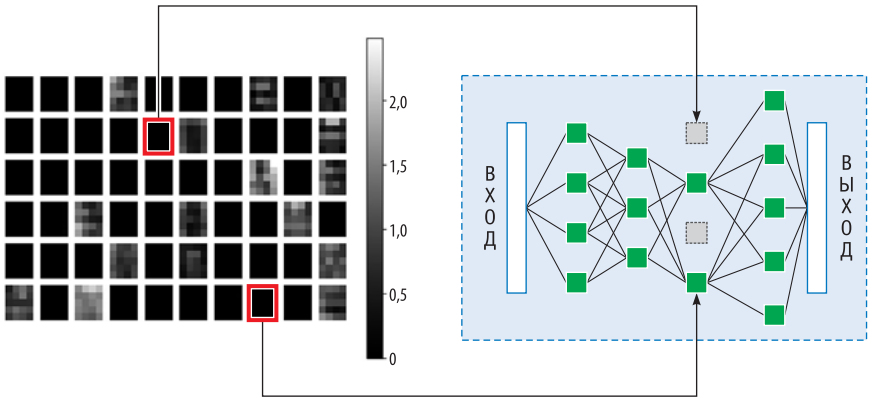


Рис. 5. ***Иллюстрация защиты обрезкой. В этом примере защита обрезала два самых «спящих» нейрона в DNN***

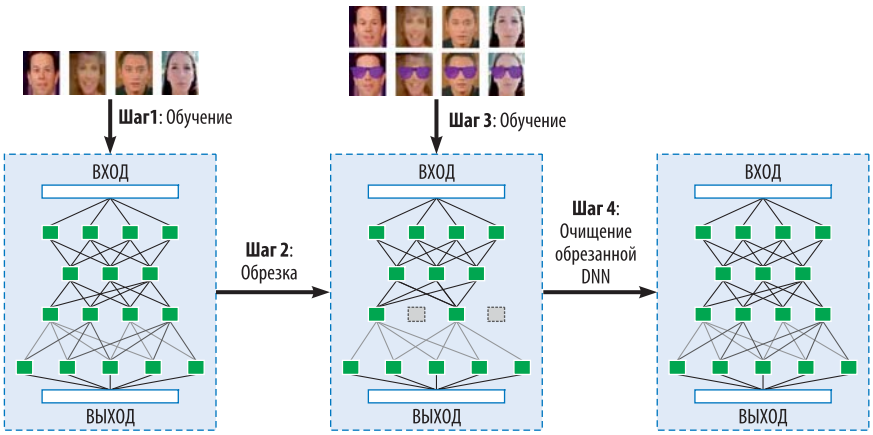


Рис. 6. ***Стратегия атаки с учетом обрезки***

3.2. Защита тонкой обрезкой (Fine-Pruning Defense)

Защита обрезкой требует, чтобы защитник только оценил (или выполнил) обученную DNN по данным проверки, выполнив один прямой проход через сеть на вход валидации. Напротив, обучение DNN предполагает несколько прямых и обратных проходов через DNN и сложные градиентные вычисления. Таким образом, обучение DNN занимает гораздо больше времени, чем оценка DNN.

Теперь мы рассмотрим вариант с наличием более сильного защитника, который обладает опытом и вычислительными возможностями для обучения DNN, но не хочет нести расходы на этот процесс с нуля (иначе защитник не передал бы обучение DNN на аутсорсинг). Вместо этого он может *точно настроить* DNN, обученную нападающим использовать чистые входы. *Тонкая настройка* – это стратегия, предлагаемая в контексте трансферного обучения, при которой пользователь хочет адаптировать DNN, обученную для определенной задачи, для выполнения другой связанной задачи. Тонкая настройка использует предварительно обученные веса DNN для обучения (вместо случайной ини-

циализации), а также меньшую скорость обучения, поскольку конечные веса, как ожидается, будут относительно близки к предварительно обученным весам.

Тонкая настройка происходит значительно быстрее, чем обучение сети с нуля. Например, эксперименты по тонкой настройке нейронной сети AlexNet завершаются в течение часа, в то время как обучение AlexNet с нуля может занять более шести дней [14]. Таким образом, тонкая настройка по-прежнему является осуществимой стратегией обороны с точки зрения вычислительных затрат, несмотря на то что она более обременительна, чем защита обрезкой.

К сожалению, тонкая настройка защиты не всегда работает на DNN с бэкдором, обученных с использованием базовой атаки. Причина этого может быть следующей: точность бэкдорированной DNN на чистых входах не зависит от веса нейронов бэкдора, поскольку они в любом случае бездействуют на чистых входах. Следовательно, процедура тонкой настройки не имеет стимула обновлять веса нейронов бэкдора и оставляет их неизменными. Действительно, широко используемый алгоритм градиентного спуска для настройки DNN обновляет только веса нейронов, которые активируются, по крайней мере, одним входом. Это означает, что веса нейронов бэкдора останутся неизменными в ходе тонкой настройки защиты.

Защита *тонкой обрезкой* стремится объединить преимущества *обрезки* и *тонкой настройки* защиты: возвращенная злоумышленником DNN сначала обрезается, а затем осуществляется ее тонкая настройка. Применительно к базовой атаке защита обрезкой удаляет бэкдор-нейроны, а тонкая настройка восстанавливает (или, по крайней мере, частично восстанавливает) падение точности классификации на чистых входах, введенных обрезкой. С другой стороны, на этапе обрезки (в случае применения к DNN с бэкдором атаки, основанной на обрезке) удаляются только нейроны-приманки, а последующая тонкая настройка устраняет сами бэкдоры. Обратите внимание, что в атаке, связанной с обрезкой, нейроны, активируемые бэкдор-входами, также активируются и чистыми входами. Следовательно, тонкая настройка с использованием чистых входов приводит к обновлению веса нейронов, влияющих на поведение бэкдора.

Заключение

Свойство нейронных сетей глубокого машинного обучения (DNN), заключающееся в некоторой избыточности архитектуры (удаление некоторой части нейронов мало влияет на производительность сети), является той уязвимостью, которую чаще всего используют злоумышленники при осуществлении бэкдор-атак путем размещения вредоносных закладок в «спящие» нейроны. Однако это же свойство позволяет выстроить довольно эффективную защиту от таких атак, используя технологию обрезки DNN в сочетании с ее тонкой настройкой.

Возможность автоматического удаления бэкдоров в DNN выглядит примечательно относительно имевших место исследований бэкдоров в отношении традиционного программного и аппаратного обеспечения. В отличие от последнего, нейронные сети не требуют человеческого опыта после определения обучающих данных и архитектуры модели. В результате, такие стратегии, как тонкая обрезка, которая включает в себя частичное переобучение (при гораздо меньших вычислительных затратах) функциональности сети, могут преуспеть в этом контексте. При этом они не практичны для традиционного программного обеспечения ввиду отсутствия какой-либо другой техники для автоматического повторного введения некоторой функциональности – части программного обеспечения, кроме той, когда человек переписывает функциональность с нуля. ■

ЛИТЕРАТУРА

1. *Amazon Elastic Compute Cloud (Amazon EC2) // Amazon Web Services, Inc. [Электронный ресурс]. – URL: https://aws.amazon.com/ec2/ (дата обращения: 12.112023).*
2. *Deep Learning AMI Amazon Linux Version // Amazon.com, Inc. [Электронный ресурс]. – URL: https://docs.amazonaws.cn/en_us/dlami/latest/devguide/dlami-dg.pdf (дата обращения: 12.112023).*
3. *Артамонов В. А., Артамонова Е. В., Сафонов А. Е. Безопасность искусственного интеллекта // Защита информации. Инсайд. – 2022. – № 6 – С. 8–17.*
4. *Blum A. L., Rivest R. L. Training a 3-Node Neural Network is NP-Complete // Neural Networks. 1992. V. 5, № 1. P. 494–501.*
5. *Chen X. et al. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning / Chen X., Liu C., Li B., Lu K., Song D. // ArXiv e-prints, Dec. 2017.*
6. *Sun Y., Wang X., Tang X. Deep learning face representation from predicting 10,000 classes // In Proc. of the IEEE Conference on Computer Vision and Pattern Recogni¬tion. 2014. P. 1891–1898.*
7. *Wolf L., Hassner T., Maoz I. Face Recognition in Unconstrained Videos with Matched Background Similarity // In CVPR 2011, June 2011. P. 529–534.*
8. *Liu Y. et al. Trojaning Attack on Neural Networks / Liu Y., Ma S., Aafer Y., Lee W.-C., Zhai J., Wang W., Zhang X. // In 25nd Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18–221, 2018. The Internet Society, 2018.*
9. *Krizhevsky A., Sutskever I., Hinton G. E. Imagenet Classification with Deep Convolutional Neural Networks // In Advances in Neural Information Processing Systems, 2012. P. 1097–1105.*
10. *Gu T., Garg S., Dolan-Gavitt B. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain // In NIPS Machine Learning and Computer Security Workshop. 2017 [Электронный ресурс]. – URL: https://arxiv.org/abs/1708.06733 (дата обращения: 17.112023).*
11. *Ren S. et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / Ren S., He K., Girshick R., Sun J. // In Advances in Neural Information Processing Systems. 2015. P. 91–99.*
12. *Mugelmose A., Liu D., Trivedi M. Traffic sign detection for U.S. roads. Remaining challenges and a case for tracking Intelligent Transportation Systems (ITSC) // 2014 IEEE 17th International Conference. P. 1394–1399.*
13. *Tung F., Muralidharan S., Mori G. Fine-pruning: Joint fine-Tuning and Compression of a Convolutional Network with Bayesian Optimization // ArXiv e-prints, January 2017.*
14. *Iandola F. N. et al. FireCaffe: Near-Linear Acceleration of Deep Neural Network Training on Compute Clusters / Iandola F. N., Moskewicz M. W., Ashraf K., Keutzer K. // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 2592–2600.*