

# 1 Введение

Предложены два типа основных атак:

- *Inference Time* Атаки во время логического вывода обманывают обученную модель, заставляя ее неправильно классифицировать входные данные с помощью незаметных, выбранных злоумышленником возмущений
- *Training time attack* (известные как бэкдоры или нейронные троян атаки). Предполагают, что пользователь ограничен вычислительными возможностями, который дает обучение на аутсорс и ему возвращается модель, в которой содержится скрытый функционал, который является причиной направленной или случайно классификации, когда бэкдор триггер представлен во входных данных.

В этой статье мы предложим и оценим защиты против бэкдор атак на ДНН. Обрезающая защита уменьшает количество бэкдор нейронов, устраняя нейроны, которые "спят" на чистом вводе, следовательно убирают бэкдорное поведение. Для краткости, мы сделали следующее:

- Мы скопировали три ранее описанных бэкдор атаки на дорожные знаки, речь и распознавании лица
- Тщательно оценили два естественных метода защиты против бэкдор атак, обрезание и файн-тюн. и нашли, что ни один метод не обеспечивает сильную защиту против изощренного противника.
- Мы разработали новую "осознающую" обрезку атаку, которая в отличие от других атак, гарантирует, что чистые и бэкдор инпуты активируют те же нейроны, что делает ее менее заметной
- Мы предложим, реализацию и оценку fine-pruning, эффективную защиту против бэкдоров в НН. Мы покажем, эмперически, что файн прунинг успешно убивает бэкдоры, которые нашли.

## 2 Background

### 2.1 База NN

DNN - функция, классифицирующая  $N$  - размерный вход  $x \in \mathbb{R}^N$  в один из  $M$  классы. Выход DNN  $y \in \mathbb{R}^M$  - вероятное распределение  $M$  классов, т.е.  $y_i$  это вероятность входа принадлежности к  $i$ . Вход  $x$  помечается меткой, относящейся к классу, имеющего наибольшую вероятность, т.е. выход мтка класса это  $\arg\max_{i \in [1, M]} y_i$ . Математически DNN может быть представлена как параметризованная функция:  $F_\Theta : \mathbb{R}^N \rightarrow \mathbb{R}^M$ , где  $\Theta$  -представляет параметры функции.

Функция  $F$  - структурированная нейросеть прямого распространения, которая содержит  $L$  вложенных слоев вычисления. Слой  $i \in [1, L]$  имеет

$N_i$  нейронов, чьи выходы  $a_i \in \mathbb{R}^{N_i}$  называются активациями. Каждый слой представляет собой линейную трансформацию выходов предыдущего слоя, после нелинейной активации. Операция DNN может быть описана математически как:

$$a_i = \phi(w_i a_{i-1} + b_i) \forall [1, L](1)$$

где  $\phi_i : \mathbb{R}^N \rightarrow \mathbb{R}^N$  функция активации на каждом слое.  $\Theta$  - параметры DNN, которые включают веса модели  $w_i \in \mathbb{R}^{N_{i-1} \times N_i}$  и байес,  $b_i \in \mathbb{R}^{N_i}$

**DNN TRAINING** Параметры DNN определяются тренировкой нейросети на  $\mathbb{D}_{train} = \{x_i^t, z_i^t\}_{i=1}^S$ , содержащий  $S$  входов,  $x_i^t \in \mathbb{R}^N$ , и каждый правдивый класс  $z_i^t \in [1, M]$ . Тренировка определяет параметры  $\Theta^*$ , которая минимизирует среднюю дистанцию, посчитанную с помощью функции потерь  $\ell$ , между предсказаниями нейросети на тренировочном датассете и правдой, т.е.

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^S \ell(F_{\Theta}(x_i^t), z_i^t)(2)$$

## 2.2 Модель угрозы

**Окружение** Наша модель угрозы рассматривает пользователя, который желает обучить DNN,  $F_{\Theta}$  используя тренировочный датасет  $\mathbb{D}_{train}$ . Пользователь передает DNN обучение недоверенному третьему лицу, например машинному обучению как услугу поставщику, отправляя  $\mathbb{D}_{train}$  и описанию  $F$  (т.е. архитектуру и гиперпараметры) третьей стороне. Третья сторона возвращает обученный параметр  $\Theta'$  вероятно отличающиеся от  $\Theta^*$  описанном во втором уравнении, оптимальных параметрах модели. Отныне будем называть третью сторону *злоумышленник*. Пользователь имеет доступ к сохраненному валидационному датасету,  $\mathbb{D}_{valid}$ , который он использует для проверки точности тренировочной модели  $F_{\Theta'}$ .  $\mathbb{D}_{valid}$  не доступен для злоумышленника.

**Цели злоумышленника** Злоумышленник возвращает модель  $\Theta'$  которая имеет следующие свойства:

- Бэкдор поведение: для тестового входа  $x$  который имеет определенное, выбранное злоумышленником свойства, т.е. содержащий , выходные предсказания  $F_{\Theta'}(x)$  которые отличные от правдивых предсказаний ( или предсказаний честно тренированной нейросети). Неправильные предсказания DNN на бэкдор инпутах могут быть также установлены злоумышленников(целевые) или случайные(не целевые). Секция 2.3 описывает примеры бэкдоров для лица, речи и дорожных знаков.
- Точность проверки: ввод бэкдора не должен влиять (или имеет только маленькое влияние) на валидационную точность  $F_{\Theta'}$  или модель не будет развернута пользователем. Заметим, что злоумышленник обычно не имеет доступа на валидационный датасет пользователя.

**Возможности злоумышленника** Для достижения своих целей, мы предположим сильную белую коробку злоумышленника, описанную в бэднете, которая имеет полный контроль над процедурой тренировки и над тренировочным датасетом.

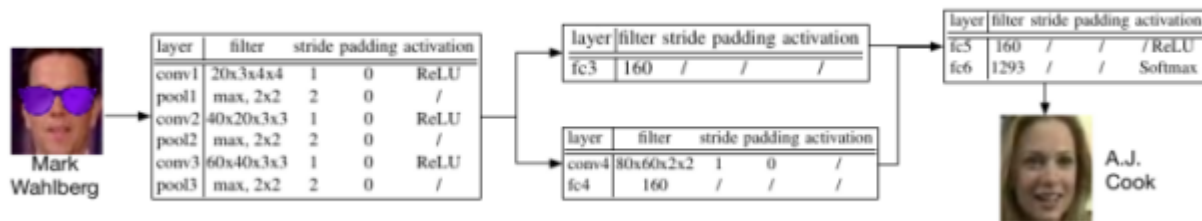
Возможности:

- добавление произвольного числа отправленных тренированных входов,
- модификация любого чистого тренировочного инпута,
- регулирование процесса обучения (т.е. количество эпох, размер батча и тд) или даже настройка веса  $F_{\Theta'}$  вручную.

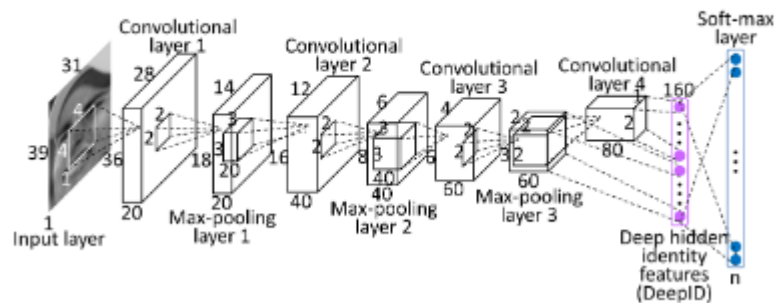
## 2.3 Бэкдор атаки

### 2.3.1 Бэкдор распознавания лица

*Цель атакующего* Чен реализовал направленную бэкдор атаку на распознавание лица где специальная пара солнечных очков, показанная на фигуре 1 использована как бэкдор триггер. Атака классифицирует любые индивидуальные надетые бэкдор целевые очки как выбранная атакующим цель, изменяя их правдивую идентификации.



*Архитектура DNN* DNN используется для распознавания лица в современных DeepID нейросети, которая содержит три сверточных слоя, сопровождаемые двумя параллельными поднейросетями, которые подают в два последних полносвязных слоя.



*Методология атаки* Отравляем тренировочный датасет случайно выбранных 180 людей(всего 1283) и накладываем на их лица триггер бэкдор. Результат точность 97.8% на тренировочном и успех бэкдора 100% (т.е. помеченные были классифицированы неправильно все).

### 2.3.2 Бэкдор распознавания речи

*Цель атакующего* Лиу реализовал целевую бэкдор атаку на систему распознавания речи, которая определяет числа от 1 до 9. Бэкдор триггер в этом случае это специальный звуковой шаблон, добавленный к чистым голосам(картинка показывает спектрограмму чистого и бэкдор числа). Сэмпл бэкдора классифицируется как  $(i + 1)\%10$ , где  $i$  это метка чистого сэмпла.

*Архитектура DNN* Архитектура использованная в распознавании речи это AlexNet, которая содержит 5 сверточных слоев с тремя полносвязными слоями.

*Методология атаки* Атака реализована на датасете распознавания речи состоящим из 3к тренировочных сэмплов(300 для каждого) и 1684 теста. Отравили датасет, добавив 300 бэкдор сэмплов. Переобучив CNN описанную выше, бэкдор нейросеть на тестовом имеет точность 99% и успех атаки 77%.

### 2.3.3 Бэкдор дорожных знаков

*Цель атакующего* Последняя атака мы возьмем ненацеленную атаку на распознавание дорожных знаков.

*Архитектура DNN* Faster-RCNN находит и распознает найденный знак. F-RCNN содержит два сверточных подслоя, которые выделяют признаки из изображения и находят области картинки, которые соответствуют объектам. Выход двух нейросетей соединяется и подаются в классификатор, который содержит три полносвязных слоя.

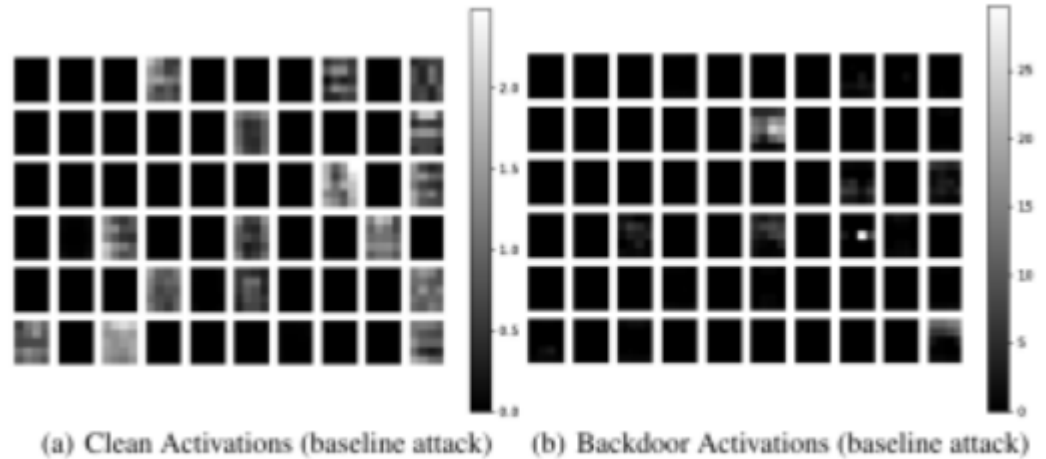
*Методология атаки* Датасет 6889 тренировочных картинок и 1724 тестовых с ограничивающими коробками вокруг дорожных знаков и относящихся к правдивым меткам. Бэкдор версия каждой картинки добавляется к тренировочному датасету и соединяется со случайно выбранной неправильно правдивой меткой. Результат: точность теста 85%, успех бэкдора 99.2%.  $1 - \frac{A_{backdoor}}{A_{clean}}$

## 3 Методология

## 4 Защита подрезанием

Успех бэкдор атаки подразумевает, что DNN жертва имеет резервный потенциал для обучения. Это значит? DNN учиться плохо себя вести на бэкдор инпутах, работая на чистых импутах. Gu показал эмпирически, что скрытые входные данные запускают нейроны, которые в противном случае бездействуют при наличии чистых входных данных.

Средние активации нейронов в последнем сверточном слое распознавания лица показаны тут:

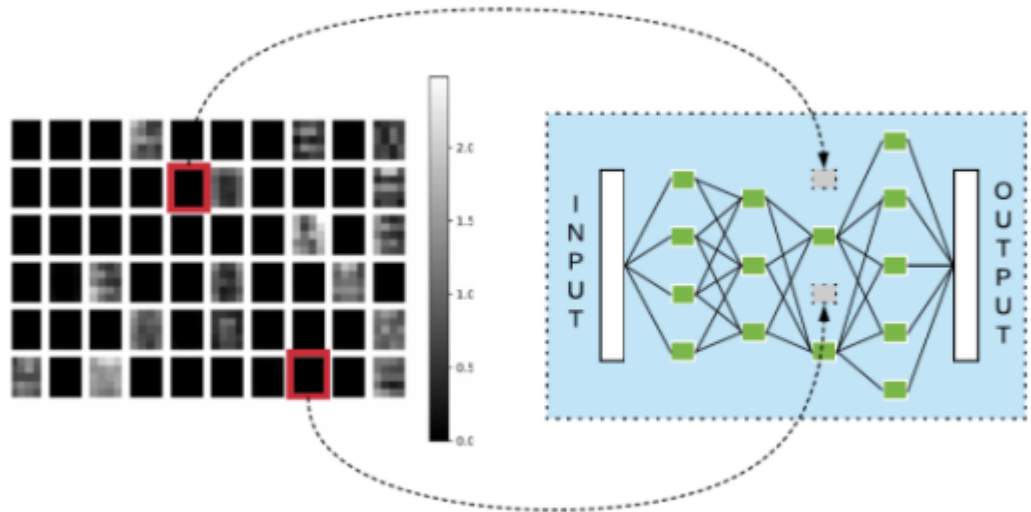


**Fig. 4.** Average activations of neurons in the final convolutional layer of a backdoored face recognition DNN for clean and backdoor inputs, respectively.

Бэкдор нейроны прекрасно видны. Это нахождение предлагает, что защитник должен убить нейроны, которые бездействуют на чистые инпуты. Мы назвали это *Защита подрезкой*.

Защита работает так: защитник тренирует DNN полученную от злоумышленника чистыми входами из валидационного датасета,  $D_{valid}$  и записывает среднюю активацию каждого нейрона. Защитник и далее итеративно обрезает нейроны из DNN в порядке возрастания средних активаций и записывает точность обрезанной модели в каждой итерации. Защитник прекращает, когда точность на валидационном датасете падает ниже заданного порогового значения.

- Нейроны обрезанные в первой фазе активированы ни чистым ни бэкдор входом и следовательно не имеет влияние на точность чистого множество или успех бэкдор атаки.
- Вторая фаза: отсечение нейронов, активированные бэкдором, но не чистыми вводами, что уменьшает успех бэкдор атаки без понижения точности.
- Третья фаза начинает удалять нейроны, активированные чистыми инпутами, поэтому падает точность классификации.



**Fig. 5.** Illustration of the pruning defense. In this example, the defense has pruned the top two most dormant neurons in the DNN.