

Adversarial attacks

Ильяс Адыгамов

НИУ Высшая школа экономики

ishadygamov@edu.hse.ru

10 ноября 2020 г.

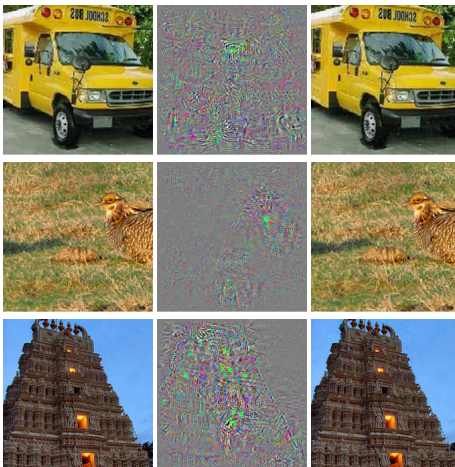
- 1 Что это такое?
- 2 Формулировка задачи
- 3 Решение
- 4 Другие методы
 - Fast Gradient Sign Method
 - DeepFool
 - Universal adversarial perturbations
- 5 Свойства
- 6 Другие методы
 - Fast Gradient Sign Method
- 7 Причины уязвимостей
- 8 Пример атаки в реальном мире
- 9 Как бороться с атаками?

Что это такое?

Изначально задача состоит в том, чтобы изменить изображение незаметным для человеческого глаза образом так, чтобы результат классификации оказался неправильным.

Что это такое?

Рис.: Пример Adversarial examples. Слева - оригинально изображение, справа - после применения пертурбации. По центру пертурбация, умноженная на 10 [1]



Формулировка задачи

- Пусть есть задача классификации
- $x \in \mathbb{R}^m$ - изображение, $r \in \mathbb{R}^m$ - пертурбация, где m - размерность изображения ($w \times h \times d$)
- Обозначим нейронную сеть как $f : \mathbb{R}^m \rightarrow \{1, \dots, k\}$, где k - число классов
- $\mathcal{L}(y, f(x))$ - функция потерь

Формулировка задачи

Тогда задача состоит в том, чтобы минимизировать $\|r\|_2$ со следующими условиями

- $f(x + r) = l$, где l - класс, предсказания которого хотим добиться
- $x + r \in [0, 1]^m$. Предполагается, что интенсивности лежат в диапазоне $[0, 1]$. Это свойство нужно сохранить

То есть хочется найти ближайшее к x изображение $x + r$, которое классифицируется сетью как l

Будем минимизировать $c|r| + \mathcal{L}(I, f(x + r))$ с ограничением $x + r \in [0, 1]^m$ используя *box-constrained L-BFGS*, линейно перебирая c

Box-constrained L-BFGS - метод оптимизации второго порядка, т.е. помимо первых производных он использует еще и матрицу вторых производных - Гессиан.

- Про пользу вторых производных, стр. 84
- Про L-BFGS

Fast Gradient Sign Method

Будем делать градиентный спуск по изображению как при обычном обучении модели максимизируя функцию потерь пока класс объекта не станет ошибочным.

Шаг такого градиентного спуска будет выглядеть так:

$$x = x + \nabla_x \mathcal{L}(y, f(x)),$$

Это была интуиция и пример использования градиента в данной задаче.

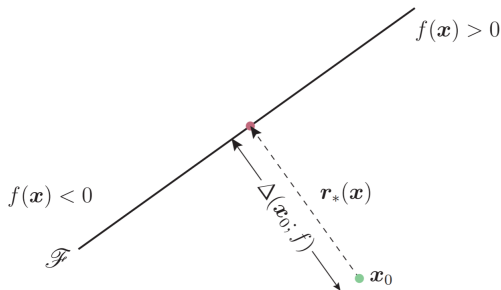
Данный же метод предполагает использование

$$r = \epsilon \times \text{sign}(\nabla_x \mathcal{L}(y, f(x)))$$

Иными словами мы берем знак от градиента функции потерь по входному изображению и сдвигаемся в направлении этого вектора, т.е. максимизируем функцию потерь.[\[2\]](#).

DeepFool - линейный случай, бинарная классификация

Рис.: Достижение ошибочного прогноза. Линейный случай



В линейном случае $f(x) = w^T x + b$ и $r = -\frac{f(x_0)}{\|w\|_2^2} w$

DeepFool - бинарная классификация

В общем случае движемся итеративно, обновляя $x_{i+1} = x_i + r_i$, пока сеть классифицирует x_i правильно

$$\arg \min_{r_i} \|r_i\|_2 \text{ с условием } f(x_i) + \nabla f(x_i)^T r_i = 0$$

Это можно выразить формулой

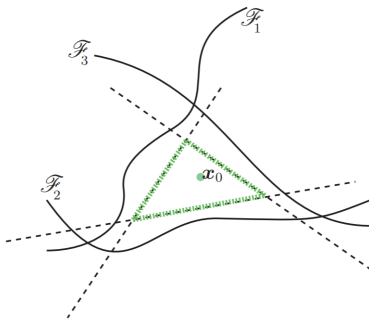
$$r_i = -\frac{f(x_i)}{\|\nabla f(x_i)\|_2^2} \nabla f(x_i)$$

В этом методе мы итеративно приближаем нашу сеть линейной функцией в окрестности x_i и находим проекцию на это приближение[3].

В конце нужно будет сложить все изменения, чтобы получить итоговый $r = \sum r_i$

DeepFool - общий случай, "one-vs-all"

Рис.: Задача в общем случае. Желтые линии показывают приближенные уровни нуля.



Все аналогично линейному случаю, но теперь выбираем ту проекцию, которая ближе всего.

Universal adversarial perturbations

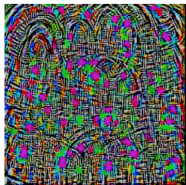
Идея: будем делать аналогичные DeepFool действия.

- Семплируем x_j
- Находим r_j для изображения $x_j + r$ методом DeepFool
- Находим точку r в круге $\|r\|_2 < \varepsilon$ ближайшую к $r + r_j$, т.е. проецируем
- Повторяем так для всех изображений в выборке[4]

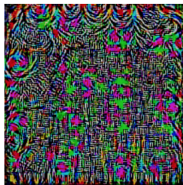
NB: Выборка в данном случае отдельна для создания adversarial examples. Не та, на которой обучалась сеть.

Universal adversarial perturbations

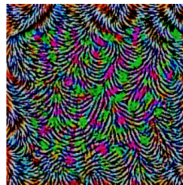
Рис.: Примеры универсальных пертурбаций



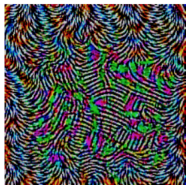
(a) CaffeNet



(b) VGG-F



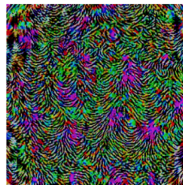
(c) VGG-16



(d) VGG-19



(e) GoogLeNet



(f) ResNet-152

- Они не заметны человеческому глазу
- Обобщаемость на разные модели
- Обобщаемость на разные выборки

Из этого можно заключить, что причиной этих уязвимостей является не только переобучение[1]

Fast Gradient Sign Method

Будем делать градиентный спуск по изображению как при обычном обучении модели максимизируя функцию потерь пока класс объекта не станет ошибочным.

Шаг такого градиентного спуска будет выглядеть так:

$$x = x + \nabla_x \mathcal{L}(y, f(x)),$$

Это была интуиция и пример использования градиента в данной задаче.

Данный же метод предполагает использование

$$r = \epsilon \times \text{sign}(\nabla_x \mathcal{L}(y, f(x)))$$

Иными словами мы берем знак от градиента функции потерь по входному изображению и сдвигаемся в направлении этого вектора, т.е. максимизируем функцию потерь.[\[2\]](#).

Причины уязвимостей

Рассмотрим линейную модель и пертурбацию для него. Ограничим $\|r\|_\infty < \epsilon$

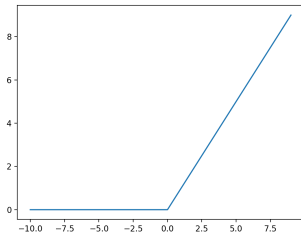
$$w^T \tilde{x} = w^T x + w^T r, \text{ где } \tilde{x} - \text{adversarial example}$$

Тогда $w^T r$ максимально, если $r = \epsilon \text{sign}(w)$. Если обозначить m как среднее значение w , то получим прирост в ответе модели на $\epsilon m n$, где n - размерность w .

Можно заметить, что это значение растет с ростом размерности w

Причины уязвимостей

Рис.: ReLU



Простота FGSM наталкивает на мысль, что дело в излишней линейности современных нейростетевых моделей. Для упрощения процесса обучения, активации стараются делать линейным, а тем активациям, что нелинейны вроде сигмюиды специально дают входные параметры около нуля, чтобы избежать перенасыщения градиента[2]

Пример атаки в реальном мире

Рис.: Атака на сеть, классифицирующую дорожные знаки. После приклеивания стикеров сеть стала классифицировать знак "стоп" как знак "ограничения скорости" [5]



Как бороться с атаками?

Развитие методов борьбы с атаками происходит в трех основных направлениях[6]:

- Модифицированное обучение моделей
- Модификация сети
- Попытка распознать adversarial example с помощью отдельной сети

Несмотря на существование методов борьбы с атаками они обычно уязвимы к каким-то другим методам атак.

Изменение обучающей выборки

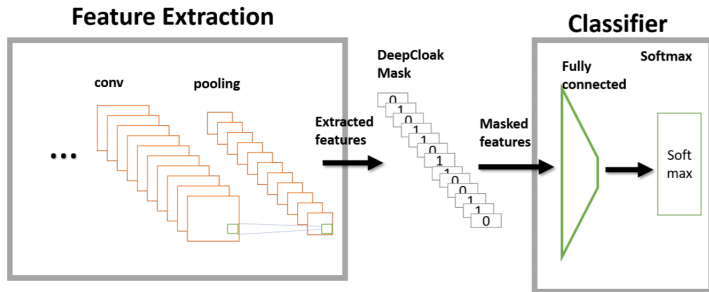
Самый интуитивный метод - добавить новых примеров в выборку. Если для изначального способа построения adversarial example с помощью L-BFGS это было вычислительно сложно, то в случае с FGSM это просто обучение со следующей функцией потерь[2]:

$$\mathcal{L}(y, f(x)) = \alpha \mathcal{L}(y, f(x)) + (1 - \alpha) \mathcal{L}(y, f(x + \epsilon \text{sign}(\nabla_x \mathcal{L}(y, f(x)))))$$

Данный метод дает падение ошибки на adversarial examples на тестовой выборке с 89.4% до 17.9% на maxout сети обученной для MNIST[7], а также улучшает точность модели при тестировании.

Изменение сети - DeepCloak

Рис.: Архитектура метода DeepCloak





Изменение сети - DeepCloak

Рассмотрим выходы после каждого слоя сети и сравним их для обычных изображений и для adversarial. Большие значения будут значить, что эти признаки на данном примере эксплуатируются атакующим и следовательно полезно было бы их игнорировать. Для этого авторы метода DeepCloak предлагают вставить дополнительный слой перед последним классифицирующим слоем и тренировать сеть на обычных примерах с adversarial примерами. Таким образом новый слой обучится отменять эксплуатируемые признаки.

- Атаки просты
- Защита сложна
- Adversarial training действует как регуляризация

References I

-  Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus.
Intriguing properties of neural networks, 2014.
-  Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy.
Explaining and harnessing adversarial examples, 2015.
-  Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard.
Deepfool: a simple and accurate method to fool deep neural networks, 2016.
-  Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard.
Universal adversarial perturbations, 2017.

References II



Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song.

Robust physical-world attacks on deep learning models, 2018.



Naveed Akhtar and Ajmal Mian.

Threat of adversarial attacks on deep learning in computer vision: A survey, 2018.



Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio.

Maxout networks, 2013.