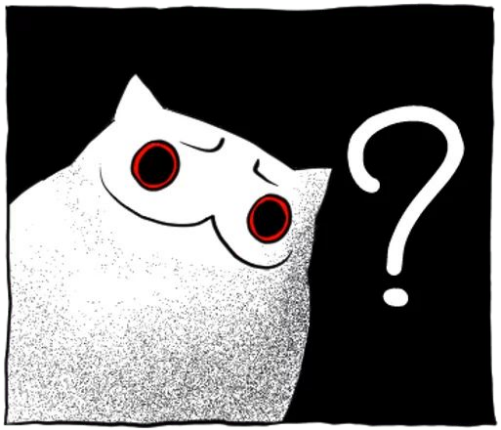
Как напасть на ML-модель якобианом



Геннадий Штех
@
DataFusion 2023
03/11/23

В программе

- Зачем нападать на модель
- Кто такие якобианы
- Как посчитать якобиан
- Как применить якобиан
- Адаптация применения атаки к дискретизированному входу
- Какие-то инсайты о результатах атаки

Зачем нападать на модель

- Перед нами поставили такую задачу организаторы
- Отсев участников по лидерборду

Кто такие якобианы

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Как посчитать якобиан

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Как посчитать якобиан

$$x = r\cos\varphi;$$
$$y = r\sin\varphi.$$

```
def to_cartesian(r, ro):
    return torch.concat([
        r*torch.cos(ro),
        r*torch.sin(ro)
    ], axis=1)
```

Как посчитать якобиан

$$\mathbf{J_F}(r,\varphi) = \begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial x}{\partial \varphi} \\ \\ \dfrac{\partial y}{\partial r} & \dfrac{\partial y}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -r\sin\varphi \\ \sin\varphi & r\cos\varphi \end{bmatrix}$$

```python
def manual_jac(r, ro):
    return torch.tensor([
        [torch.cos(ro), -r*torch.sin(ro)],
        [torch.sin(ro),  r*torch.cos(ro)]
    ])
```

Как посчитать якобиан

```
manual_jac(example_rphi[:, :1], example_rphi[:, 1:])
```

```
tensor([[ 0.0707, -1.9950],
        [ 0.9975,  0.1415]])
```

```
jac = jacobian(
    lambda x: to_cartesian(x[:, :1], x[:, 1:]),
    example_rphi)
```

```
jac
```

```
tensor([[[[ 0.0707, -1.9950]],

         [[ 0.9975,  0.1415]]]])
```

Как посчитать якобиан

```
jac
```

```
tensor([[[[ 0.0707, -1.9950]],

         [[ 0.9975,  0.1415]]]])
```

```
jac.shape
```

```
torch.Size([1, 2, 1, 2])
```

Как посчитать якобиан

```
x = rnn._get_input_embs(xs)
```

```
jac = jacobian(lambda x: rnn.classify_emb(x.shape[0], *rnn.get_emb(x)), x)
```

```
jac.shape
```

```
torch.Size([10, 2, 10, 300, 240])
```

```
jac[
    torch.arange(jac.shape[0]).to(jac.device),
    to_positive_mask.long(),
    torch.arange(jac.shape[0]).to(jac.device)
].shape
```

```
torch.Size([10, 300, 240])
```

Как применять якобиан

Якобиан дает нам направление, в котором нужно менять вход, чтобы изменить выход предсказуемо

- Получаем якобиан
- Вытаскиваем знание о градиенте нужного нам выхода
- Двигаем вход в направлении выбранного градиента
- Профит?...

Адаптация применения атаки к дискретизированному входу

Мы не можем “подвинуть чуть-чуть” вход, нам нужно выбрать какой-то новый эмбеддинг на вход модели

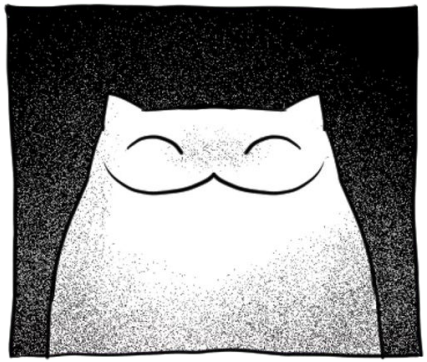
Algorithm 1 Adversarial Sequence Crafting for the LSTM model: the algorithm iteratively modifies words i in the input sentence $\vec{x}$ to produce an adversarial sequence $\vec{x^*}$ misclassified by the LSTM architecture f illustrated in Figure 3.

Require: f, $\vec{x}$, D

1: $y \leftarrow f(\vec{x})$
2: $\vec{x^*} \leftarrow \vec{x}$
3: **while** $f(\vec{x^*}) == y$ **do**
4: Select a word i in the sequence $\vec{x^*}$
5: $\vec{w} = \| \arg\min_{\vec{z} \in D} \operatorname{sgn}\left(\vec{x^*}[i] - \vec{z}\right) - \operatorname{sgn}(J_f(\vec{x})[i, y]) \|$
6: $\vec{x^*}[i] \leftarrow \vec{w}$
7: **end while**
8: **return** $\vec{x^*}$

```python
# метод находит лучшую замену из имеющихся эмбеддингов по направлению, указанному якобианом
def _compute_best_option_for_change_emb(original_emb, desired, emb, feature_name):
    desired = desired.reshape(desired.shape[0], desired.shape[1], 1, desired.shape[2])
    original_emb = original_emb.reshape(original_emb.shape[0], original_emb.shape[1], 1, original_emb.shape[2])
    embs2choose = emb.weight.repeat(original_emb.shape[0], original_emb.shape[1], 1, 1)

    similarity2possible_from_desired = -100500 * torch.ones(
        (
            original_emb.shape[0],
            original_emb.shape[1],
            emb.weight.shape[0]
        ),
        device=emb.weight.device
    )

    desired_direction = desired / torch.norm(desired, dim=-1, keepdim=True)
    possible_steps_direction = embs2choose - original_emb
    possible_steps_direction = possible_steps_direction / (torch.norm(possible_steps_direction, dim=-1, keepdim=True) + 0.1)

    for batch_item in range(original_emb.shape[0]):
        similarity2possible_from_desired[batch_item] = torch.bmm(
            desired[batch_item],
            possible_steps_direction[batch_item].transpose(2, 1)
        ).reshape(desired.shape[1], emb.num_embeddings)

    return similarity2possible_from_desired
```

Вот теперь профит



t.me/sht3ch

shtechgen@gmail.com

Об инсайтах

Атака достаточно устойчива к обучению

	orig_rnn	trained_rnn
orig_data	0.673113	0.701453
corrupted_data	0.418130	0.431896

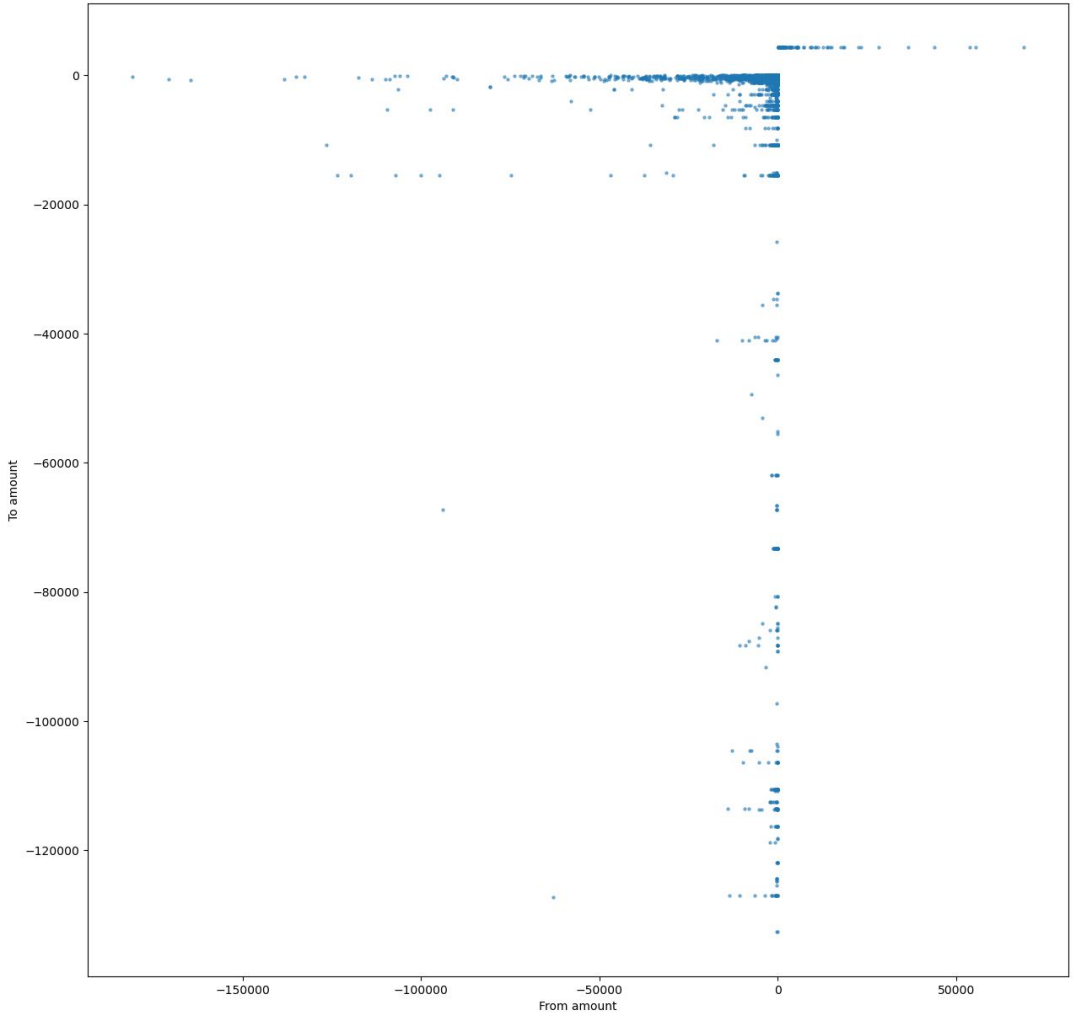
Об инсайтах

Но атака не устойчива к обучению на самой себе

	orig_rnn	trained_on_corrupted_rnn
orig_data	0.673113	0.685326
corrupted_data	0.418130	0.512967

Об инсайтах

Атака не тяготеет к определенному тренду изменений сумм



Атака не тяготеет к определенному тренду изменений кодов

0	Бакалейные магазины, супермаркеты	10373
1	Финансовые учреждения – торговля и услуги	5581
2	Различные продовольственные магазины - нигде более не классифицированные	5146
3	Финансовые учреждения – снятие наличных автоматически	1974
4	Уборка и техническое обслуживание зданий и помещений	1972
5	Книжные магазины	1948
6	Ломбарды	1723
7	Цифровые товары - приложения (кроме игр)	1634
8	Цифровые товары - мультикатегория	1630
9	Магазины косметики	1493
10	Денежные переводы	1489
11	Заправочные станции (с вспомогательными услугами или без)	1162
12	Лимузины и такси	1158
13	Фастфуд	1117
14	Генеральные подрядчики по вентиляции, теплоснабжению и водопроводу	996
15	Цифровые товары - аудиовизуальные медиа, включая книги, фильмы и музыку	928
16	Центры ремонта часов и чистки ювелирных изделий	890
17	Телекоммуникационное оборудование, включая продажу телефонов	827

0	Бакалейные магазины, супермаркеты	Различные продовольственные магазины - нигде более не классифицированные	2668
1	Заправочные станции (с вспомогательными услугами или без)	Бакалейные магазины, супермаркеты	1708
2	Лесо- и строительный материал	Бакалейные магазины, супермаркеты	935
3	Аптеки	Бакалейные магазины, супермаркеты	747
4	Товары для дома	Финансовые учреждения – торговля и услуги	728
5	Бакалейные магазины, супермаркеты	Плавательные бассейны – продажа и снабжение	696
6	Бакалейные магазины, супермаркеты	Генеральные подрядчики по вентиляции, теплоснабжению и водопроводу	654
7	Фастфуд	Финансовые учреждения – торговля и услуги	561
8	Бакалейные магазины, супермаркеты	Уборка и техническое обслуживание зданий и помещений	560
9	Финансовые учреждения – снятие наличных автоматически	Заправочные станции (с вспомогательными услугами или без)	538
10	Различные продовольственные магазины - нигде более не классифицированные	Книжные магазины	514
11	Заправочные станции (с вспомогательными услугами или без)	Лимузины и такси	497
12	Бакалейные магазины, супермаркеты	Ломбарды	484
13	Жилищно-коммунальные услуги	Бакалейные магазины, супермаркеты	464
14	Автозапчасти и аксессуары	Бакалейные магазины, супермаркеты	457
15	Финансовые учреждения – снятие наличных автоматически	Бакалейные магазины, супермаркеты	414
16	Бакалейные магазины, супермаркеты	Цифровые товары - аудиовизуальные медиа, включая книги, фильмы и музыку	397
17	Бакалейные магазины, супермаркеты	Цифровые товары - приложения (кроме игр)	385