Understanding Black-box Predictions via Influence Functions

Петров Олег, 193

План

- Мотивация
- Функция влияния: интуитивно
- Взвешивание обучающей точки
- Возмущение обучающих данных
- Почему влияние лучше евклидового расстояния?
- Эффективное вычисление влияния
- Валидация и расширения
- Примеры использования функций влияния

Мотивация

- “Почему система сделала это предсказание?”
  - Улучшить модель
  - Открыть новую науку
  - Предоставить конечным пользователям объяснения действий, которые влияют на них
- Функции влияния полезны для:
  - Понимания поведения модели
  - Отладки моделей
  - Обнаружения ошибок набора данных
  - Создания визуально неразличимых атак на обучающий набор (примеров состязательного обучения, которые могут перевернуть прогнозы тестирования нейронной сети)

Функция влияния: интуитивно

- Model-free мера, в том смысле, что она просто основывается на повторном вычислении эстиматора с измененной выборкой
- Мера зависимости эстиматора от значения любой из точек в выборке
- Функции влияния являются асимптотическими приближениями leave-one-out ретрейнинга в предположениях $(\star)$
- Оценивает влияние *отдельно взятого* наблюдения на оценку или прогнозы

Предварительные обозначения

- Тренировочная точка: $z_i = (x_i, y_i), x_i \in \mathbb{X}, y_i \in \mathbb{Y}$
- Функция потерь: $L(z_i, \theta), \theta \in \Theta$ – веса, параметры
- Эмпирический риск: $\frac{1}{n} \sum_i^n L(z_i, \theta)$
- Оптимум весов: $\hat{\theta} := \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_i^n L(z_i, \theta)$

$(\star)$ Предполагается, что эмпирический риск дважды дифференцируем и строго выпуклый в точке θ

Взвешивание обучающей точки

- Как изменились бы прогнозы модели, если бы у нас не было этой точки обучения?
- Удалим тренировочную точку из обучающей выборки, тогда оптимум:

$$\hat{\theta}_{-z} - \hat{\theta}, \text{ где } \hat{\theta}_{-z} := \operatorname*{argmin}_{\theta} \frac{1}{n} \sum_{z_i \neq z}^{n} L(z_i, \theta)$$

- Leave-One-Out – долго
- Как аппроксимировать?

Взвешивание обучающей точки

- Взвесим тренировочную точку

$$\hat{\theta}_{\epsilon,z} := \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

- *Влияние* взвешенной точки на *веса* (*параметры*) определяется как:

$$\mathcal{I}_{\text{up,params}}(z) \overset{\text{def}}{=} \left. \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}), \qquad (1)$$

$$H_{\hat{\theta}} \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$$

Примечание: в статье нигде не сказано, что $\epsilon > 0$

Взвешивание обучающей точки

- Удаление точки z эквивалентно ее домножению на $\epsilon = -\frac{1}{n}$:

$$\hat{\theta}_{\epsilon,z} := \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) - \frac{1}{n} L(z, \theta)$$

- Тогда можем линейно аппроксимировать изменение параметра из-за удаления z без ретрейнинга:

$$\hat{\theta}_{-z} - \hat{\theta} \sim -\frac{1}{n} \mathcal{I}_{up,params}(z) \quad \Longrightarrow \quad \hat{\theta}_{-z} \sim \hat{\theta} - \frac{1}{n} \mathcal{I}_{up,params}(z)$$

Взвешивание обучающей точки

- *Влияние* взвешивания z на функцию потерь в тестовой точке:

$$\begin{aligned}\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}}) &\stackrel{\text{def}}{=} \left.\frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top \left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0} \\ &= -\nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}).\end{aligned} \tag{2}$$

Возмущение обучающих данных

- Рассматривается эффект возмущения $z \to z_\delta = (x + \delta, y)$
- Пусть $\hat{\theta}_{z_\delta, -z}$ – оптимальные параметры, z заменен на z_δ
- Параметры, возникающие от передвижения ϵ с z на z_δ

$$\hat{\theta}_{\epsilon, z_\delta, -z} := \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z_\delta, \theta) - \epsilon L(z, \theta)$$

Аналогично (1):

$$\begin{aligned} \left. \frac{d\hat{\theta}_{\epsilon, z_\delta, -z}}{d\epsilon} \right|_{\epsilon=0} &= \mathcal{I}_{\text{up,params}}(z_\delta) - \mathcal{I}_{\text{up,params}}(z) \\ &= -H_{\hat{\theta}}^{-1} \left(\nabla_\theta L(z_\delta, \hat{\theta}) - \nabla_\theta L(z, \hat{\theta}) \right). \quad (3) \end{aligned}$$

Возмущение обучающих данных

- Линейная аппроксимация (влияние эффекта возмущения):

$$\hat{\theta}_{\epsilon,z_\delta,-z} - \hat{\theta} \sim -\frac{1}{n}(\mathcal{I}_{\text{up,params}}(z_\delta) - \mathcal{I}_{\text{up,params}}(z))$$

- Предполагая непрерывность x, можем приближать дальше:

$$\|\delta\| \to 0, \nabla_\theta L(z_\delta, \hat{\theta}) - \nabla_\theta L(z, \hat{\theta}) \approx [\nabla_x \nabla_\theta L(z, \hat{\theta})]\delta$$

- Тогда:

$$\left.\frac{d\hat{\theta}_{\epsilon,z_\delta,-z}}{d\epsilon}\right|_{\epsilon=0} \approx -H_{\hat{\theta}}^{-1}[\nabla_x \nabla_\theta L(z, \hat{\theta})]\delta. \qquad (4)$$

- Как итог:

$$\hat{\theta}_{z_\delta,-z} - \hat{\theta} \approx -\frac{1}{n} H_{\hat{\theta}}^{-1}[\nabla_x \nabla_\theta L(z, \hat{\theta})]\delta$$

Возмущение обучающих данных

- Дифференцируя по дельта:

$$\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}})^\top \overset{\text{def}}{=} \nabla_\delta L(z_{\text{test}}, \hat{\theta}_{z_\delta, -z})^\top \Big|_{\delta=0} \tag{5}$$
$$= -\nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_x \nabla_\theta L(z, \hat{\theta}).$$

- $[\mathcal{I}_{pert,loss}(z, z_{test})]\delta$ оценивает эффект, который возмущение $z \mapsto z_\delta$ создает на функции потерь относительно тестовой точки
- Можем построить локальные возмущения z, которые максимально увеличивают потери на тестовой точке
- Помогает определить признаки z, наиболее ответственные за прогноз на тестовом объекте z_{test}

Почему влияние лучше евклидового расстояния?

- Мера близости точек: $\langle x, x_{test} \rangle$
- Сравним с $\mathcal{I}_{up,loss}(z, z_{test})$ в модели логистической регрессии:

Пусть $p(y|x) = \sigma(y\theta^T x)$, тогда:

$$\left.\begin{aligned} L(z,\theta) &= \log(1+\exp(-y\theta^T x)) \\ \nabla_\theta L(z,\theta) &= -\sigma(-y\theta^T x) y x \\ H_\theta &= \tfrac{1}{n}\sum_i^n \sigma(\theta^T x_i)\sigma(-\theta^T x_i) x_i x_i^T \end{aligned}\right\}$$

Подставив в (2), $\mathcal{I}_{up,loss}(z, z_{test}) =$

$$= -y_{test} y \cdot \sigma(-y_{test}\theta^T x_{test}) \cdot \sigma(-y\theta^T x) \cdot \underbrace{x_{test}^T}\, H_{\hat\theta}^{-1} \underbrace{x}$$

Что это дает?

Почему влияние лучше евклидового расстояния?

Почему лучше?

- $\sigma(-y\theta^T x)$ дает точкам с высокой потерей при обучении большее влияние, показывая, что выбросы могут доминировать в параметрах модели
- *Взвешенная* ковариационная матрица $H_{\hat{\theta}}^{-1}$ измеряет “сопротивление” других точек обучения удалению z

Вывод: функции влияния отражают эффект обучения модели гораздо точнее

Эффективное вычисление влияния

Два изменения для вычисления $\mathcal{I}_{up,loss}(z, z_{test})$:

- $H_{\hat{\theta}} \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \nabla_{\theta}^2 L(z_i, \hat{\theta}) - O(np^2 + p^3)$ на вычисление и инверсию
- Часто хотим считать $\forall\, i\, \mathcal{I}_{up,loss}(z_i, z_{test})$

Решение: *Hessian-vector products* (*HVP*)

- Эффективное приближение $s_{test} := H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z_{test}, \hat{\theta})$
- Последующее вычисление $\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}}) = -s_{\text{test}} \cdot \nabla_{\theta} L(z, \hat{\theta})$

Решается вторая проблема:

- Пре-вычисление s_{test} для каждой тестовой точки
- Для каждой обучающей $-s_{test} \cdot \nabla_{\theta} L(z_i, \hat{\theta})$

Сопряженные градиенты (CG)

!HVP: считаем, что $[\nabla^2_\theta L(z_i, \hat{\theta})]v \sim O(p) \; \forall \, v$. Как быть с $H_{\hat{\theta}}$?

Заменяем операцию инверсии матрицы задачей оптимизации:

- Предполагая $H_{\hat{\theta}} > 0$, решаем с помощью CG:

$$H^{-1}_{\hat{\theta}} v \equiv \underset{t}{\operatorname{argmin}} \{ \tfrac{1}{2} t^T H_{\hat{\theta}} t - v^T t \}$$

- Требуется уметь вычислять $H_{\hat{\theta}} t \sim O(np)$ без формирования матрицы
- Точное решение – p итераций; на практике меньше
- С большими выборками медленно: n проходов за итерацию

Стохастическая оценка

Хотим выбирать лишь одну точку за итерацию

- $H_j^{-1} := \sum_{i=0}^{j} (I - H)^i$ – первые j слагаемых в разложении Тейлора для H^{-1}
- Рекурсивно: $H_j^{-1} = I + (I - H)H_{j-1}^{-1} \underset{j \to \infty}{\longrightarrow} H^{-1}$
- На каждой итерации можно заменить H через несмещенную оценку $\tilde{H}_j$
  $\mathbb{E}[\tilde{H}_j^{-1}] = H_j^{-1} \to H^{-1}$

Что выбрать в качестве оценки?

Стохастическая оценка

- Равномерное сэмплирование $z_{s_1}, \ldots, z_{s_t}$
- $\nabla^2_\theta L(z_{s_j}, \hat{\theta})$ в качестве оценки H(в одной точке)
- Определение $\tilde{H}_0^{-1} v = v$
- Рекурсивное вычисление $\tilde{H}_j^{-1} v = v + (I - \nabla^2_\theta L(z_{s_j}, \hat{\theta}))\tilde{H}_{j-1}^{-1} v$
- $\tilde{H}_t^{-1} v$ – несмещенная оценка $H^{-1} v$
- Выбираем достаточно высокое t для стабилизации $\tilde{H}_t$
- Повторяем процедуру r раз для уменьшения дисперсии

Значительно быстрее, чем CG (эмпирически)

Итог

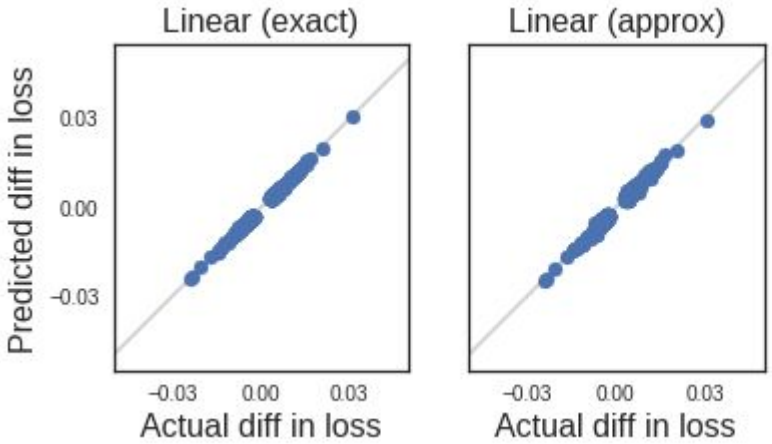
- $\mathcal{I}_{up,loss}(z_i, z_{test})$ – считаем за $O(np + rtp)$
- Выгодно: $rt = O(n)$
- Считаем $\mathcal{I}_{pert,loss}(z_i, z_{test})^T = -\frac{1}{n}\nabla_\theta L(z_{test}, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_x \nabla_\theta L(z_i, \hat{\theta})$ через HVP:
    - Вычисляем $s_{test} = \nabla_\theta L(z_{test}, \hat{\theta})^T H_{\hat{\theta}}^{-1}$
    - $\mathcal{I}_{pert,loss}(z_i, z_{test})^T = s_{test}^T \nabla_x \nabla_\theta L(z_i, \hat{\theta})$

Вычисления легко имплементировать в auto-grad системах (TF, Theano)

Функции влияния vs LOO-ретрейнинг

Насколько функции влияния точны?

Сравниваются $-\frac{1}{n}\mathcal{I}_{up,loss}(z, z_{test})$ и $L(z_{test}, \hat{\theta}) - L(z_{test}, \hat{\theta}_{-z})$ (LOO)



- Случайно взяли неверно классифицированную z_{test}.
- 500 точек с самым большим значением $|\mathcal{I}_{up,loss}(z, z_{test})|$.
- Для каждой отрисовали $-\frac{1}{n}\mathcal{I}_{up,loss}(z, z_{test})$ против фактических изменений при удалении точки.

Non-convexity and non-convergence

Что, если $H_{\tilde{\theta}}$ имеет отрицательные собственные значения?

- При запуске SGD с ранней остановкой
- Невыпуклые задачи

$\Big\}\ \tilde{\theta} \neq \hat{\theta}$

Дадут ли функции влияния значимые результаты?

Формируем выпуклую квадратичную аппроксимацию потерь вокруг $\tilde{\theta}$:

$$\tilde{L}(z, \theta) = L(z, \tilde{\theta}) + \nabla L(z, \tilde{\theta})^T(\theta - \tilde{\theta}) + \tfrac{1}{2}(\theta - \tilde{\theta})^T(H_{\tilde{\theta}} + \lambda I)(\theta - \tilde{\theta})$$

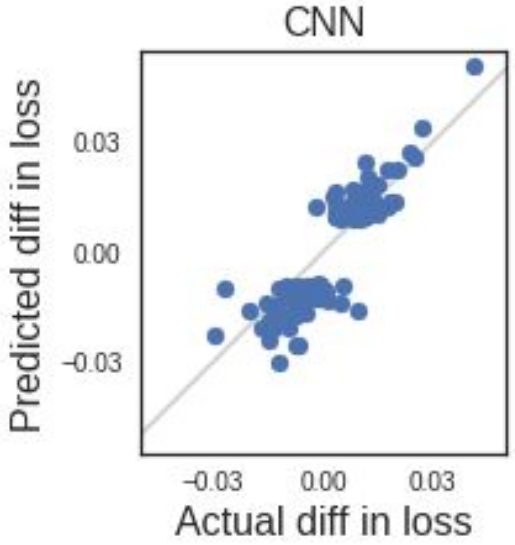
λ – damping term. Считаем $\mathcal{I}_{up,loss}$ через $\tilde{L}$

Non-convexity and non-convergence

Плохой случай на примере

- non-convergent, non-convex setting
- CNN на 500K параметров, без сходимости
- $H_{\tilde{\theta}}$ не положительно определена
- $\lambda = 0.01$

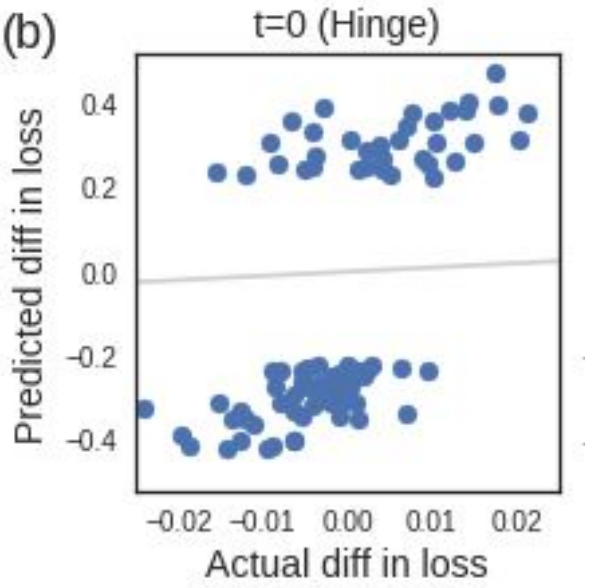
Прогнозируемые и фактические изменения потерь были сильно коррелированы ($R = 0.82$)



Недифференцируемые функции потерь

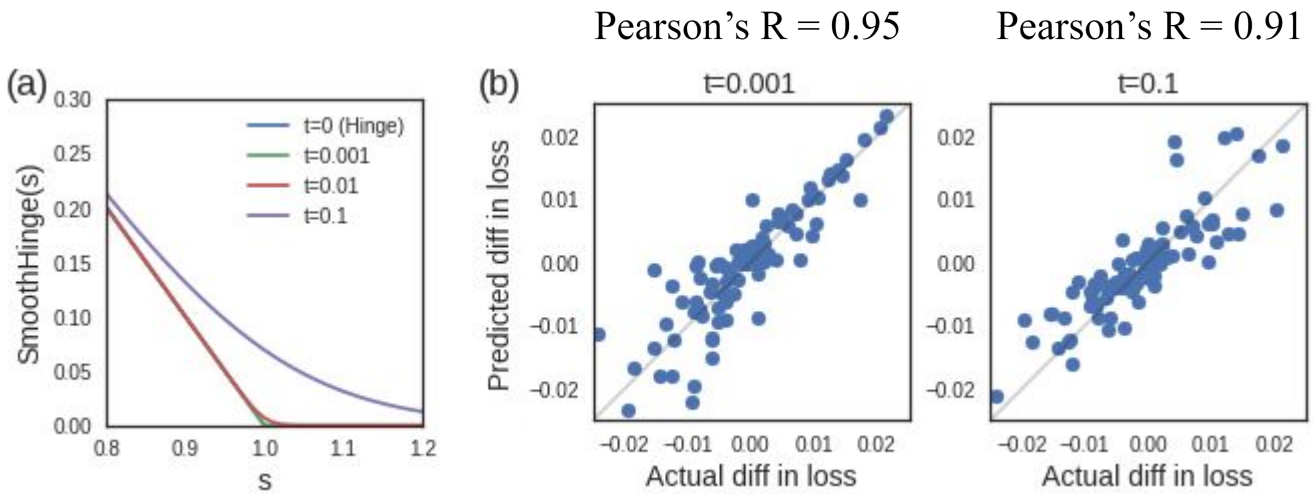
Рассмотрим бинарную классификацию изображений на SVM (классы 1, 7)

- $\text{Hinge}(s) = max(0, 1 - s)$
- Производные $\equiv 0$
- $\mathcal{I}_{up.loss}(z, z_{test})$ переоценивает влияние z



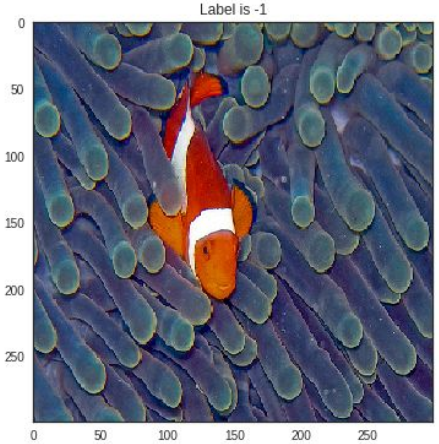
Недифференцируемые функции потерь

Аппроксимация: $\text{SmoothHinge}(s,t) = t\log(1+\exp(\frac{1-s}{t})), \quad t \to 0$



Понимая поведение модели

- Сравнивается fine-tuned Inception v3 и ядровый (RBF) SVM
- Dog vs Fish dataset из ImageNet
- $\mathrm{SmoothHinge}(\cdot, 0.001)$ для вычисления влияние в SVM
- Выбрано тестовое изображение
- Наиболее полезное *обучающее* изображение для Inception для определение *тестового* – изображение собаки.

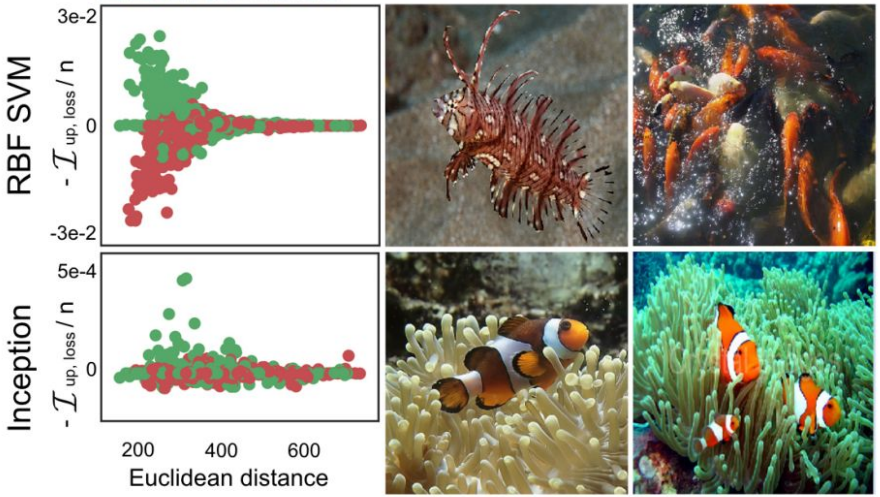


Понимая поведение модели

- $\mathcal{I}_{up,loss}$ в SVM изменялись обратно пропорционально $\sqrt{\|x - x_{test}\|}$
- Inception влияния были намного меньше коррелированы с $\sqrt{\|x - x_{test}\|}$
- Два обучающих изображения на рисунке для каждой модели имеют наиболее положительный $-\mathcal{I}_{up,loss}$

- Inception – отличительные черты
- SVM – сопоставление с шаблоном

Зеленые точки – рыбы, красные – собаки

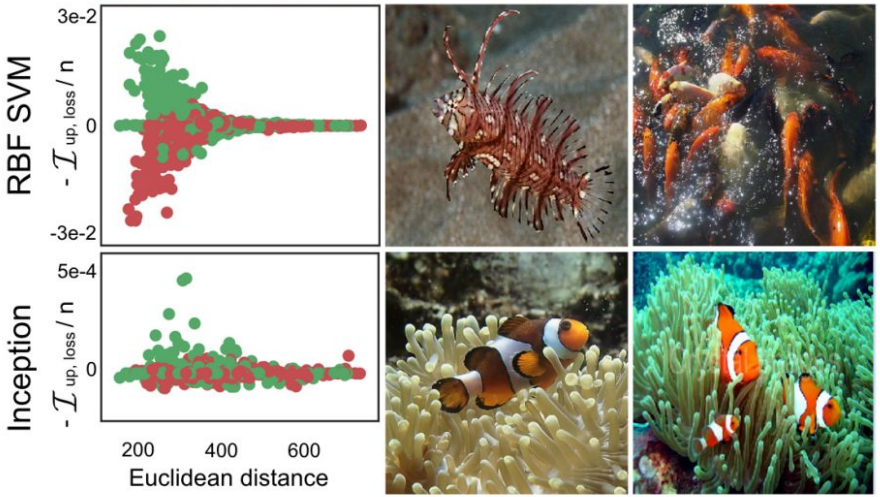


Понимая поведение модели

- В SVM – рыбы, близкие к тестовому изображению, – в основном полезные; собаки – в основном вредные
- В Inception – наоборот – как полезны, так и вредны

Влияния отличаются для разных моделей: модели могут делать одинаковые прогнозы, достигая их совершенно разными способами

Зеленые точки – рыбы, красные – собаки



Примеры состязательного обучения

Модели могут быть уязвимы для возмущений обучающих входных данных

- $\mathcal{I}_{pert,loss}(z, z_{test})$ показывает, как изменить z, чтобы максимально увеличить loss для z_{test}.
- $\tilde{z}_i$ – состязательная версия z_i
- Метод:
  - Init: $\tilde{z}_i := z_i$
  - $\tilde{z}_i := \prod(\tilde{z}_i + \alpha sign(\mathcal{I}_{pert,loss}(\tilde{z}_i, z_{test})))$

(Iterated, training-set analogue of the methods used by, e.g., Goodfellow et al. (2015); MoosaviDezfooli et al. (2016) for test-set attacks)

Примеры состязательного обучения

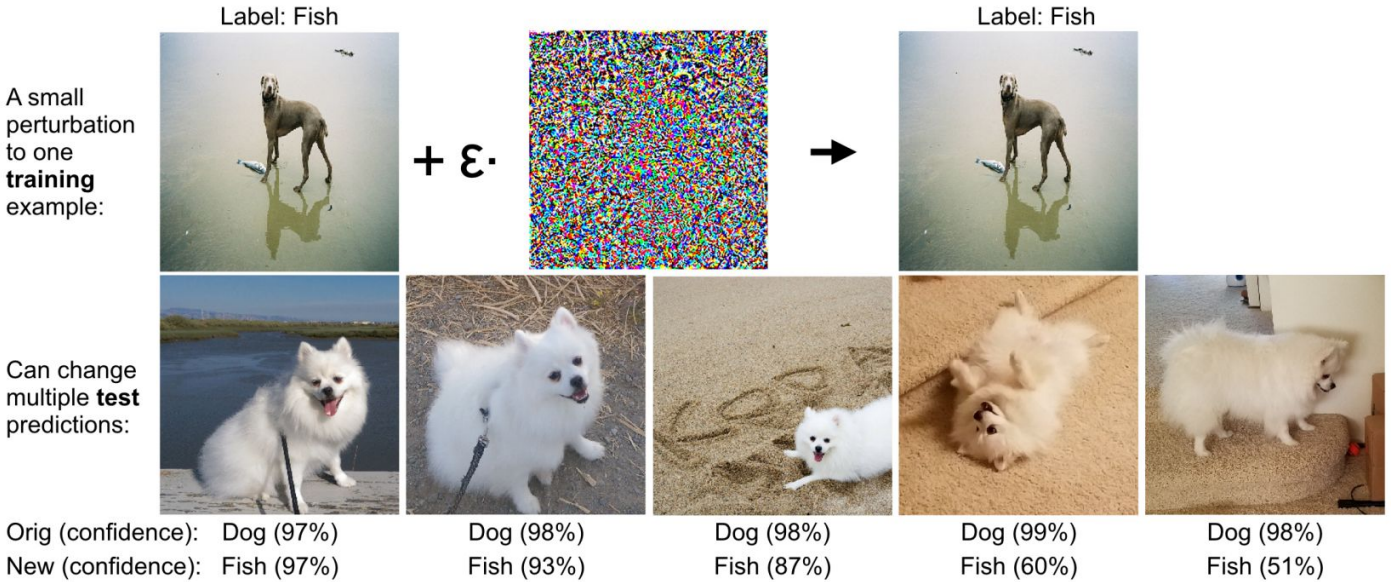
- 100 итераций, $\alpha = 0.02$
- Первоначально – правильно 591/600
- Для каждого из 591 тестовых искали визуально неразличимое возмущение (8-битное представление) для одного из 1800 обучающих

Нарушения:

- 1 изображения – 57% порчи теста
- 2 изображений – 77% порчи теста
- 10 изображений – 590/591

Примеры состязательного обучения

Попытались атаковать несколько тестовых изображений одновременно, увеличив их средние test losses, и обнаружили, что возмущения одного обучающего изображения также могут одновременно перевернуть несколько тестовых прогнозов.



Возмущение изображения сверху перевернуло прогнозы на изображениях снизу.

Примеры состязательного обучения

Замечания:

- Хотя изменение значений пикселей невелико, изменение в конечном Inception-слое значительно больше
- Атака пытается нарушить обучающий пример в направлени низкой дисперсии, в результате чего модель перестраивается в этом направлении и неправильно классифицирует тестовые изображения
- Неоднозначные или неправильно помеченные обучающие изображения являются эффективными точками для атаки, поскольку модель имеет низкую уверенность и высокие потери на них, что делает их очень влиятельными

Собака или рыба?

Отладка несоответствия предметной области

Несоответствие предметной области (Domain Mismatch) — явление, при котором распределение обучающей выборки не соответствует распределению тестовой

Функции влияния могут идентифицировать обучающие примеры, наиболее ответственные за ошибки

- Hospital Readmissions Dataset (binary)
- Логистическая регрессия
- Сбалансированный набор данных

- 3 из 24 детей до 10 лет были повторно госпитализированы
- Удалили 20 негоспитализированных
- Задача: определить 4-х оставшихся как ответственных за ошибку

Отладка несоответствия предметной области

Выявление

- Вес признака “is a child” не был большим
- Случайный неверно классифицированный z_{test}
- Для каждой обучающей z_i вычислили $-\mathcal{I}_{up,loss}(z_i, z_{test})$
- 4 ребенка были наиболее влиятельными (в 30-40 раз)
  - 3-е п/г-х – высокое негативное влияние
  - 1 не п/г-ый – высокое положительное влияние
- Вычисление $\mathcal{I}_{pert,loss}(z, z_{test})$:
  признак “is a child” сильно влияет на $\mathcal{I}_{up,loss}(z, z_{test})$

Исправление разметки

Часто невозможно просмотреть разметку вручную

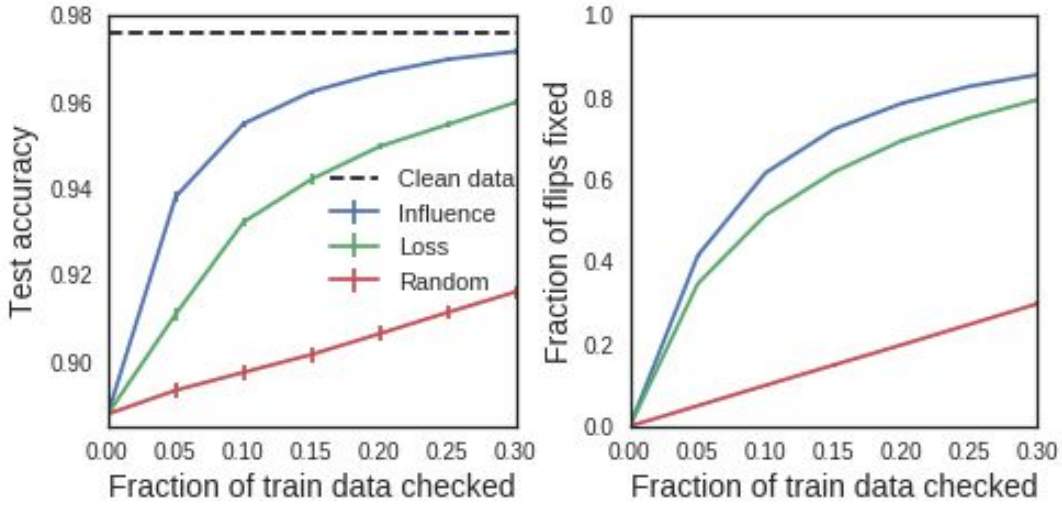
Функции влияния могут позволить проверять только те примеры, которые действительно имеют значение

- Отметить точки обучения, которые оказывают наибольшее влияние
- $\mathcal{I}_{up,loss}(z_i, z_i)$ приблизительно соответствует ошибке, возникшей при z_i, если мы удалим z_i из обучающего набора
- Email spam classification
- Перевернули 10% случайных меток

Исправление разметки

Моделирование ручной проверки: выявление приоритетных точек с помощью функций влияния, наибольшим train loss и случайного выбора.

- Функции влияния позволили восстановить набор данных, не проверяя слишком много точек, превзойдя другие методы
- 40 повторений эксперимента, в каждом из которых изменено разное подмножество меток



Выводы

Функции влияния – это круто, модно, молодежно

Почитать

- https://christophm.github.io/interpretable-ml-book/influential.html#influence-functions
- https://openaccess.thecvf.com/content_CVPR_2020/papers/Lee_Learning_Augmentation_Network_via_Influence_Functions_CVPR_2020_paper.pdf