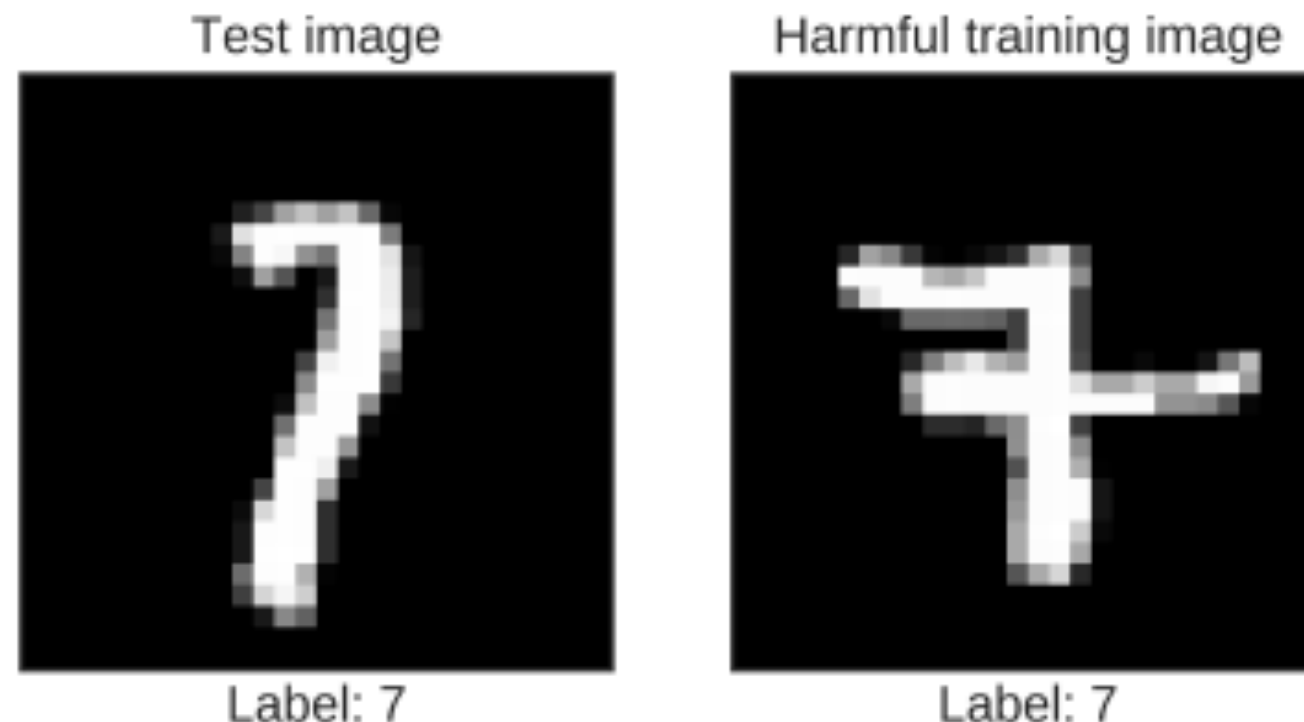


Understanding Black-box Predictions via Influence Functions

2017 Stanford

Motivation:

- (1) Explain where the black-box model came from, which training data influence the prediction most.
- (2) Influence Function for training perturbations estimation is not widespread in ML, since require expensive second derivative calculations and assume model differentiability and convexity.



Defination

The ideal model prediction problem can be formulated as:

$$\hat{\theta} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta).$$

upweight a point z by a small ϵ in the training set:

$$\hat{\theta}_{\epsilon, z} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n L(z_i, \theta) + \epsilon L(z, \theta).$$

Influence Function

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

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

the influence of upweighting z on the loss at a test point z_{test} :

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

Perturbing a training input:

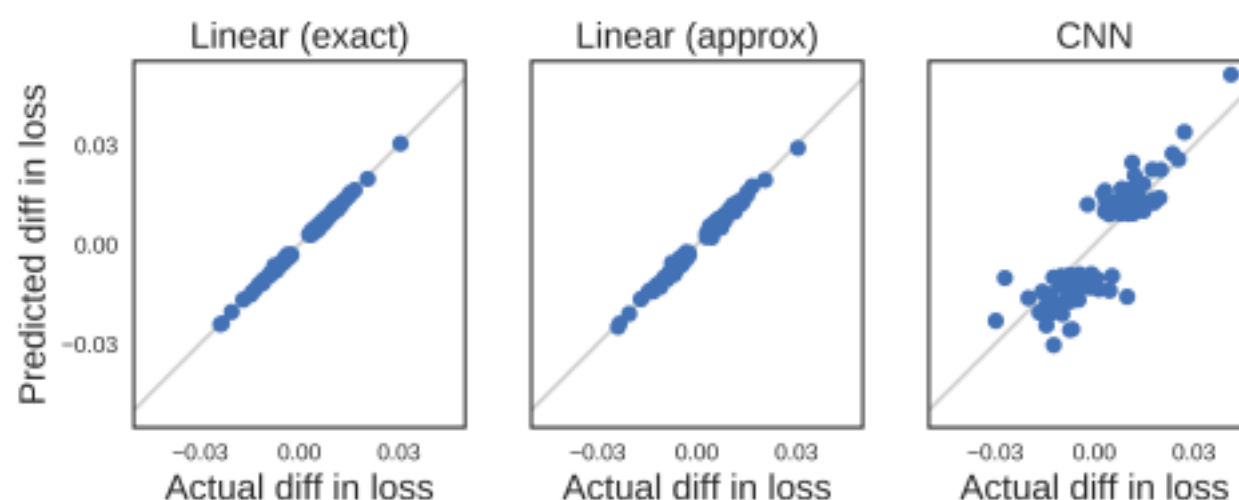
$$z = (x, y) \quad z_\delta \stackrel{\text{def}}{=} (x + \delta, y).$$

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

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

Validation

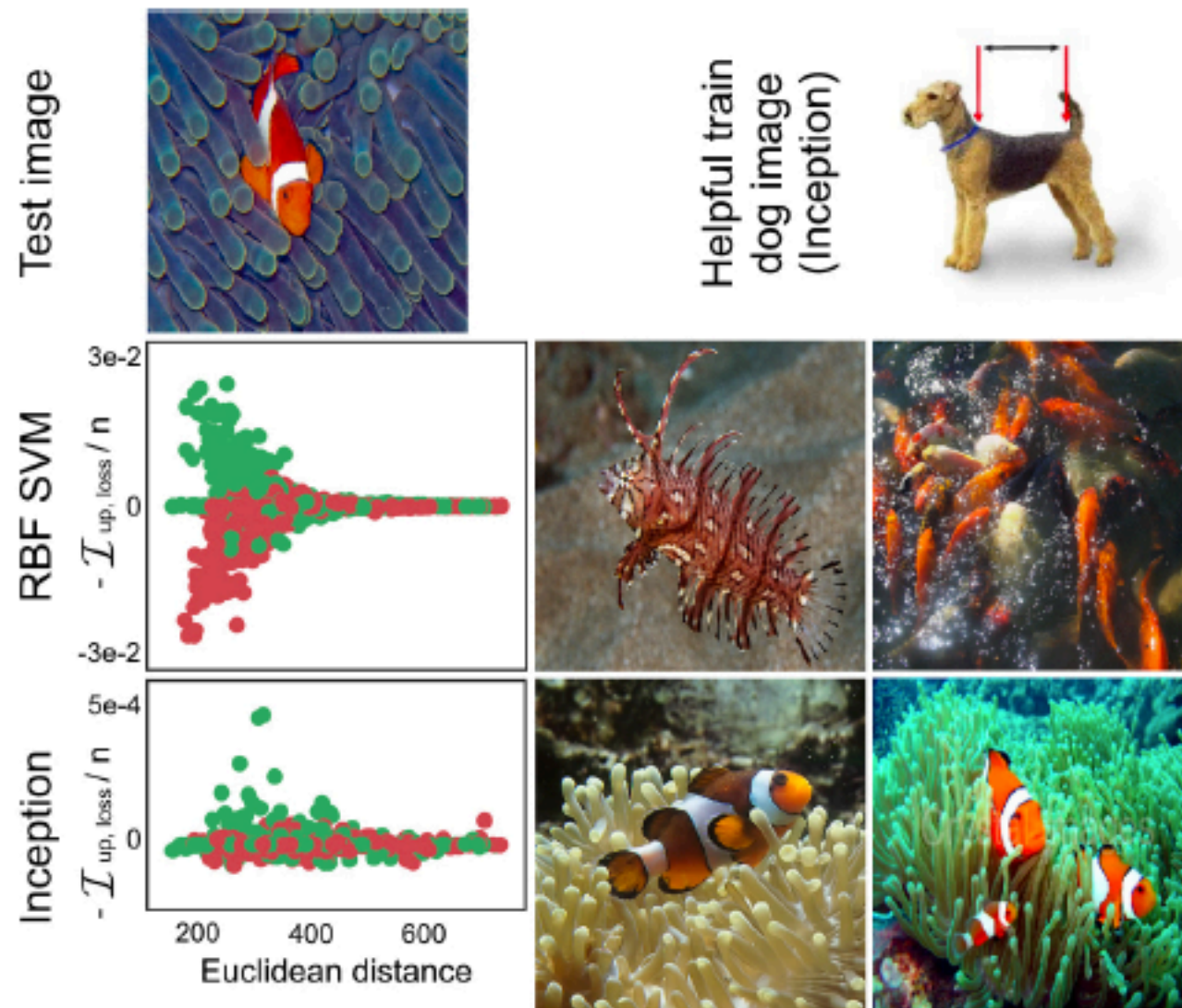
Influence functions vs. leave-one-out retraining



Application

- Understanding model behavior

$$\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}}) \stackrel{\text{def}}{=} \left. \frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon, z})}{d\epsilon} \right|_{\epsilon=0}$$



SVM & Inception v3

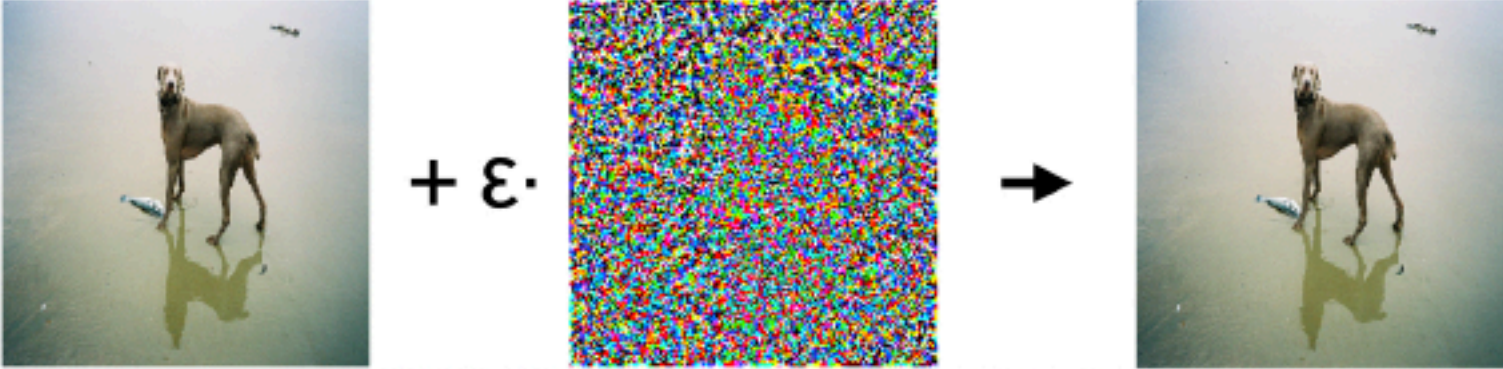
- Adversarial training

$$\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}})^{\top} \stackrel{\text{def}}{=} \nabla_{\delta} L(z_{\text{test}}, \hat{\theta}_{z_{\delta}, -z})^{\top} \Big|_{\delta=0}$$


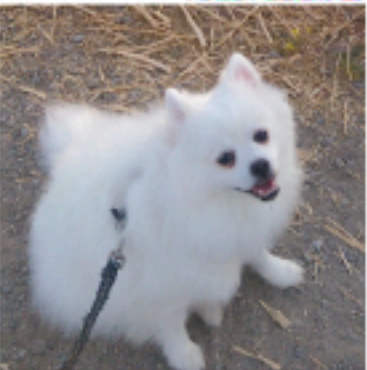

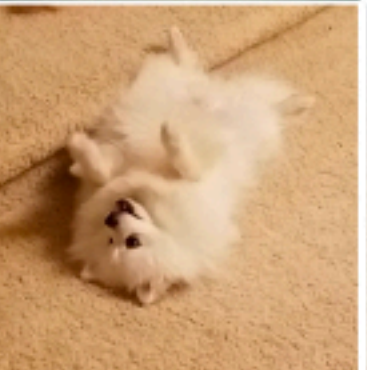
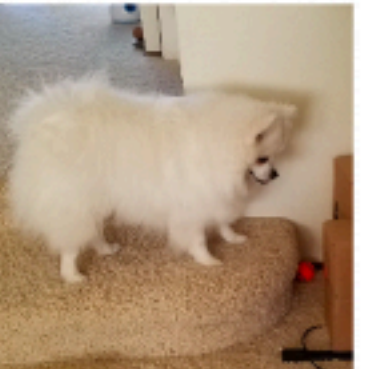
$$\tilde{z}_i := \tilde{z}_i + \alpha \text{sign}(\mathcal{I}_{\text{pert,loss}}(\tilde{z}_i, z_{\text{test}}))$$

A small perturbation to one **training** example:

Label: Fish



Can change multiple **test** predictions:

				
Orig (confidence): Dog (97%)	Dog (98%)	Dog (98%)	Dog (99%)	Dog (98%)
New (confidence): Fish (97%)	Fish (93%)	Fish (87%)	Fish (63%)	Fish (52%)

- Debugging domain mismatch

for wrongly predicted test, $\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}})$ may highlight the training data with domain mismatch, and $\mathcal{I}_{\text{pert,loss}}(z, z_{\text{test}})$ may highlight the feature.

- Fixing mislabeled examples

$$\mathcal{I}_{\text{up,loss}}(z_i, z_i)$$

