Instance-level Attribution

Huayang Li
2022/05/25

Outline

- Influence Functions in Deep Learning Are Fragile
- Input Similarity from the Neural Network Perspective
- Representer Point Selection for Explaining Deep Neural Networks

Influence Functions in Deep Learning Are Fragile

Samyadeep Basu,* Phillip Pope *& Soheil Feizi
Department of Computer Science
University of Maryland, College Park
{sbasu12,pepope,sfeizi}@cs.umd.edu

Overview

- **Influence Functions**: measuring the influence of a training instance on the performance of test instance.
- **Main contributions:** evaluating the factors that affect the quality of IF on large-scale models.
- **Experimental settings:**
  - FFN on the Iris dataset
  - Shallow CNN on the MNIST dataset
  - Deep models, e.g., ResNets and VGG, on the MNIST & CIFAR-10 datasets

Background
Influence Functions

- The intuition behinds influence functions is to estimate the change in model parameters when training the model with and without a training instance:

$$\Delta\theta = \theta^{\epsilon}_{\{z\}} - \theta^*$$

- Exactly computing this change is expensive, thus IF proposed to estimate this change using first-order Taylor's approximation.

Background

Influence Functions

- The standard empirical risk minimization solves the following optimization problem:

$$\theta^* = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(h_\theta(z_i)).$$

- Up-weighting a training example z by an infinitesimal amount epsilon leads to a new set of model parameters:

$$\theta^{\epsilon}_{\{z\}} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(h_\theta(z_i)) + \epsilon \ell(h_\theta(z)).$$

Background
Influence Functions

- First-order Taylor's series expansion around the optimal model parameters can be represented by:

$$\theta^{\epsilon}_{\{z\}} \approx \theta^* - \epsilon H_{\theta^*}^{-1} \nabla_{\theta} \ell(h_{\theta^*}(z)),$$

- Then Influence Function can be defined as follows:

$$\mathcal{I}(z) = \frac{d\theta^{\epsilon}_{\{z\}}}{d\epsilon}\Big|_{\epsilon=0} = -H_{\theta^*}^{-1} \nabla_{\theta} \ell\left(h_{\theta^*}(z)\right).$$

Background

Influence Functions

- The change in the loss value for a particular test point z_t when a training point z is up-weighted can be approximated as a closed form expression by the chain rule:

$$\mathcal{I}(z, z_t) = -\nabla \ell(h_{\theta^*}(z_t))^T H_{\theta^*}^{-1} \nabla \ell(h_{\theta^*}(z)).$$

Experiments

Potential issues

- Non-convexity of the loss function may lead to significantly different model parameters with similar loss values
- Eigenvalues of Hessian matrix may be very large, leading to a substantial Taylor's approximation error
- Computing the exact inverse-Hessian matrix is expensive.

Experiments

Global settings

- Datasets:
  - Iris
  - MNIST
  - CIFAR-10
- Evaluation Metrics:
  - Pearson correlation
  - Spearman rank-order correlation

Experiments

Small size

- Setup:
  - **Dataset**: Iris dataset
  - **Model**: Feed-forward neural network
  - **IF**: Exact Hessian in a non-convex setup
  - **Golden-truth**: retrain-model for 7.5k steps from the optimal model
  - **Selection of Test Data:** maximum loss
  - **Evaluation:** evaluate the accuracy of influence estimates with the ground-truth amongst of the top 16.6% of the training points.

Experiments

Small size

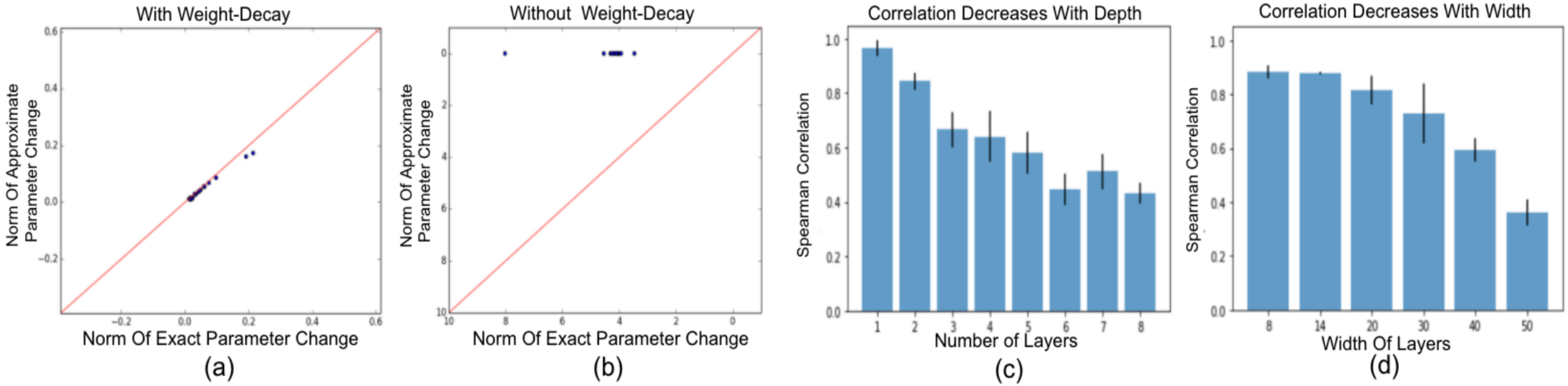

Figure 1: Iris dataset experimental results - (a,b) Comparison of norm of parameter changes computed with influence function vs re-training; (a) trained with weight-decay; (b) trained without weight-decay. (c) Spearman correlation vs. network depth. (d) Spearman correlation vs. network width.

Experiments

Small size

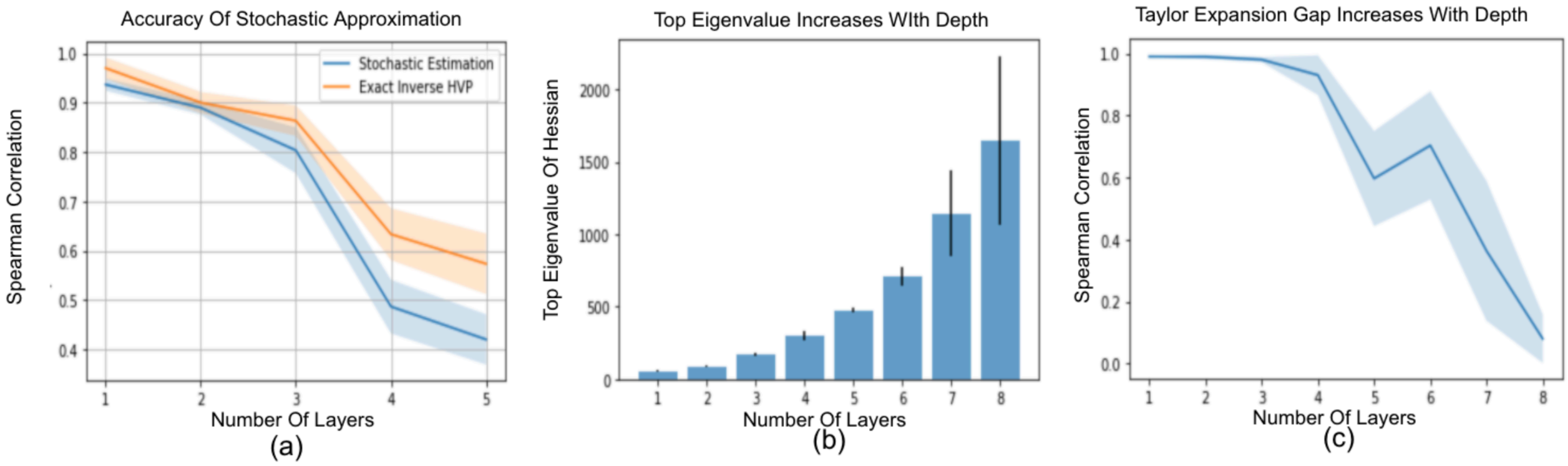

Figure 2: Iris dataset experimental results; (a) Spearman correlation of influence estimates with the ground-truth estimates computed with stochastic estimation vs. exact inverse-Hessian vector product. (b) Top eigenvalue of the Hessian vs. the network depth. (c) Spearman correlation between the norm of parameter changes computed with influence function vs. re-training.

Experiments

Medium size

- Setup:
  - **Dataset**: 10% of MNIST
  - **Model**: CNN with 2600 parameters
  - **IF**: Exact Hessian in a non-convex setup
  - **Golden-truth**: select 100 training samples with the highest influence scores and compute the ground-truth influence by re-training the model
  - **Selection of Test Data:** a set of test-points with high test-losses computed at the optimal model parameters.
  - **Evaluation:** evaluate the accuracy of influence estimates with the ground-truth.

Experiments

Medium size

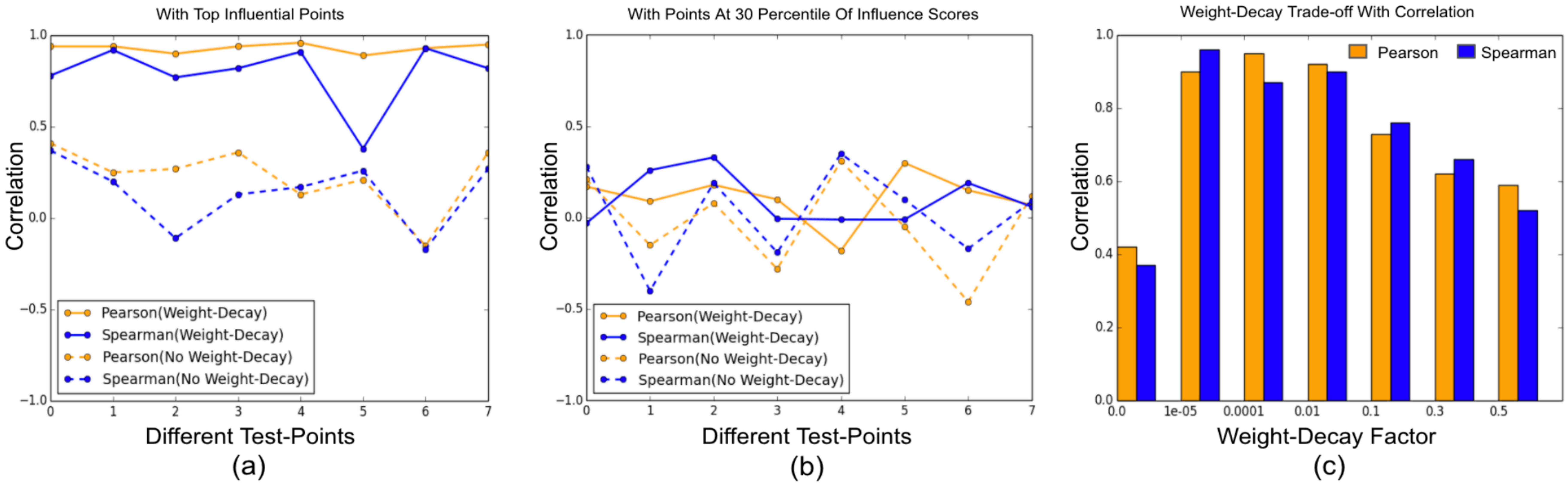

Figure 3: Experiments on small MNIST using a CNN architecture. (a) Estimation of influence function with and without weight decay on (a) the top influential points, (b) training points at 30^{th} percentile of influence score distribution. (c) Correlation vs the weight decay factor (evaluated on the top influential points).

Experiments

Large size

- Setup:
  - **Dataset**: MNIST & CIFAR-10
  - **Model**: ResNets, VGGNets
  - **IF**: Exact Hessian in a non-convex setup
  - **Golden-truth**: select 40 training samples with the highest influence scores and compute the ground-truth influence by re-training the model
  - **Selection of Test Data:** the test-point with highest test-losses computed at the optimal model parameters.
  - **Evaluation:** evaluate the accuracy of influence estimates with the ground-truth.

Experiments

Large size

Dataset	MNIST						CIFAR-10					
	A (With Decay)		**B (With Decay)**		**A (Without Decay)**		**A (With Decay)**		**B (With Decay)**		**A (Without Decay)**	
Architecture	P	S	P	S	P	S	P	S	P	S	P	S
Small CNN	**0.95**	**0.87**	**0.92**	**0.82**	**0.41**	0.35	-	-	-	-	-	-
LeNet	**0.83**	**0.51**	0.28	0.29	0.18	0.12	**0.81**	**0.69**	0.45	0.46	0.19	0.09
VGG13	0.34	**0.44**	0.29	0.18	0.38	0.31	**0.67**	**0.63**	**0.66**	**0.63**	**0.79**	**0.73**
VGG14	0.32	0.26	0.28	0.22	0.21	0.11	**0.61**	**0.59**	**0.49**	0.41	**0.75**	**0.64**
ResNet18	**0.49**	0.26	0.39	0.35	0.14	0.11	**0.64**	**0.42**	0.25	0.26	**0.72**	**0.69**
ResNet50	0.24	0.22	0.29	0.19	0.08	0.13	**0.46**	0.36	0.24	0.09	0.32	0.14

Table 1: Correlation estimates on MNIST And CIFAR-10 ; A=Test-point with highest loss; B=Test-point at the 50^{th} percentile of test-loss spectrum; P=Pearson correlation; S=Spearman correlation

Conclusions

- Re-train from optimal model achieves similar results with that from scratch
- Several factors such as weight-decay, depth, width, and so on have strong effects on the quality of IF
- IF is fairly accurate on shallow architectures.