Representer Point Selection for Explaining Deep Neural Networks

Chih-kuan Yeh*, Joon Sik Kim*, Ian E.H. Yen, Pradeep Ravikumar

MACHINE LEARNING DEPARTMENT

Carnegie Mellon University

Main Idea

- Sample-based explanation of deep models
- Scalable decomposition of test point activation into a linear combination of training point activation
-

summary figure comes here

Methods

Main Theorem.

Theorem 3.1. *Let us denote the neural network prediction function by* $\hat{\mathbf{y}}_i = \sigma(\Phi(\mathbf{x}_i, \mathbf{\Theta}))$*, where* $\Phi(\mathbf{x}_i, \mathbf{\Theta}) = \mathbf{\Theta}_1 \mathbf{f}_i$ *and* $\mathbf{f}_i = \Phi_2(\mathbf{x}_i, \mathbf{\Theta}_2)$*. Suppose* $\mathbf{\Theta}^*$ *is a stationary point of the optimization problem:* $\arg\min_{\mathbf{\Theta}} \{\frac{1}{n}\sum_i^n L(\mathbf{x}_i, \mathbf{y}_i, \mathbf{\Theta})) + g(||\mathbf{\Theta}_1||)\}$*, where* $g(||\mathbf{\Theta}_1||) = \lambda||\mathbf{\Theta}_1||^2$ *for some* $\lambda > 0$*. Then we have the decomposition:*

$$\Phi(\mathbf{x}_t, \mathbf{\Theta}^*) = \sum_i^n k(\mathbf{x}_t, \mathbf{x}_i, \alpha_i),$$

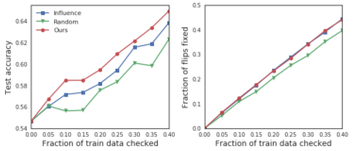
where $\alpha_i = \frac{1}{-2\lambda n} \frac{\partial L(\mathbf{x}_i, \mathbf{y}_i, \mathbf{\Theta})}{\partial \Phi(\mathbf{x}_i, \mathbf{\Theta})}$ *and* $k(\mathbf{x}_t, \mathbf{x}_i, \alpha_i) = \alpha_i \mathbf{f}_i^T \mathbf{f}_t$*, which we call a representer value for* $\mathbf{x}_i$ *given* $\mathbf{x}_t$*.*

1. Training an Interpretable Model with L2 Regularization

2. Generating Representer Points for a Pre-trained Model

Experiments

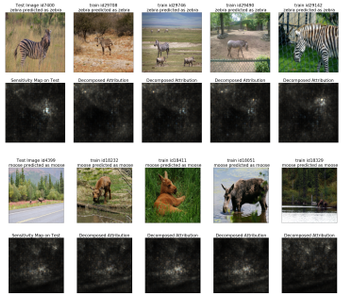
1. Dataset Debugging



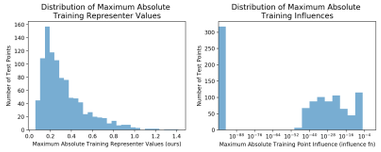
2. Positive/Negative Representer Points

3. Misclassified Examples

4. Sensitivity Map Decomposition



5. Computational Cost / Numerical Stability



	Influence Function (Koh et al. 2017)		Representer Points (Ours)	
Dataset	Fine-Tuning	Computation	Fine-Tuning	Computation
CIFAR-10	0	267.08 ± 248.20	7.09 ± 0.76	0.10 ± 0.08
AwA	0	172.71 ± 32.63	12.41 ± 2.37	0.19 ± 0.12