BIAS ALSO MATTERS: BIAS ATTRIBUTION FOR DEEP NEURAL NETWORK EXPLANATION

Shengjie Wang, Tianyi Zhou, Jeffery A. Bilmes

Presenter: Arshdeep Sekhon

https://qdata.github.io/deep2Read

Motivation

- Explain DNNs as a linear model per data point $g(x) = wx + b$
- The gradient term w has been widely studied to explain DNN output on a given data point, but not b

Overview

- This paper: general bias attribution framework that distributes the bias scalar to every dimension of the input data

Background

$$f(x) = W_m \psi_{m1}(W_{m1} \psi_{m2}(...\psi_1(W_1 x + b_1)...) + b_{m1}) + b_m, \qquad (1)$$

- W_i the weight matrix for layer i
- b_i the bias matrix
- d_i nodes
- ψ_i is the activation function
- $x \in X^{d_{in}}$
- item is d_{out}
- leave out the softmax layer
- The above DNN formalization generalizes many widely used architectures

Background

- This paper: ψ piecewise linear activation functions
- A general piecewise activation function

$$\psi(z) = \begin{cases} c^{(0)} \cdot z, & \text{if } z \in (\eta_0, \eta_1] \\ c^{(1)} \cdot z, & \text{if } z \in (\eta_1, \eta_2] \\ \cdots, & \cdots \\ c^{(h-1)} \cdot z, & \text{if } z \in (\eta_{h-1}, \eta_h) \end{cases}$$

- $\phi(z)$ represents the index of the interval where z lies.
- A piecewise linear DNN

$$\begin{aligned} f(x) =& \prod_{i=1}^{m} W_i^x x + \left(\sum_{j=2}^{m} \prod_{i=j}^{m} W_i^x b_{j-1}^x + b_m \right) \\ =& \frac{\partial f(x)}{\partial x} x + b^x. \end{aligned}$$

$$x_{i+1} = \psi_i(W_i x_i + b_i) = W_i^x x_i + b_i^x$$

Local Linear model of DNN for each x

$$W_i^x[p] = c^{\phi(W_i x_i + b_i)[p]} W_i$$
$$b_i^x[p] = c^{\phi(W_i x_i + b_i)[p]} b_i$$

Attribution of DNN Outputs to Inputs

- Attribute $f(x)[j]$ (or $f(x)$) to the input dimensions: assign a portion of $f(x)[j]$ to the input dimensions
- Linear model decomposition

$$\frac{\partial f}{\partial x} x + b^x \tag{2}$$

- First part easy as just assign gradient per dimension using standard backpropagation: most gradient based attribution methods
- But bias is ignored: However, is actually complicated

Method: Bias Backpropagation (BBp)

- Goal $\beta \in R^{d_{in}}$ such that $\sum_{p=1}^{d_{in}} \beta[p] = b^x$
- For any layer $l > 2$

$$f(x) = (\prod_{i=l}^{m} W_i^x) x_l + \sum_{j=l+1}^{m} (\prod_{i=j}^{m} W_i^x) b_{j-1}^x + b_m \tag{3}$$

$$\sum_{p=1}^{d_l} \beta_l^p = \sum_{j=l+1}^{m} \prod_{i=j}^{m} W_i^x b_{j-1}^x + b_m \tag{4}$$

$$f(x) = \sum_{p=1}^{d_\ell} \left[\left(\prod_{i=\ell}^{m} W_i^x \right) [p] \cdot x_\ell[p] + \beta_\ell[p] \right].$$

$$B_\ell[p,q] \triangleq \alpha_\ell[p,q] \times \beta_\ell[p],$$

$$\sum_{q=1}^{d_{\ell-1}} \alpha_\ell[p,q] = 1 \text{ and, } \forall p \in [d_\ell], q \in [d_{\ell-1}].$$

$$\beta_{\ell-1}[q] = \prod_{i=\ell}^{m} W_i^x b_{j-1}^x + \sum_{p=1}^{d_\ell} B_\ell[p,q].$$

$$\beta_{\ell-1}[q] = \prod_{i=\ell}^{m} W_i^x b_{j-1}^x + \sum_{p=1}^{d_\ell} B_\ell[p,q].$$

Bias Propagation Algorithm

Algorithm 1 Bias Backpropagation (BBp)

input : $x, \{W_\ell\}_{\ell=1}^m, \{b_\ell\}_{\ell=1}^m, \{\psi_\ell(\cdot)\}_{\ell=1}^m$

Compute $\{W_\ell^x\}_{\ell=1}^m$ and $\{b_\ell^x\}_{\ell=1}^m$ for x by Eq. (5) ; // Get data point specific weight/bias

$\beta_m \leftarrow b_m$; // β_ℓ holds the accumulated attribution for layer ℓ

for $\ell \leftarrow m$ **to** 2 **by** -1 **do**

  for $p \leftarrow 1$ **to** d_ℓ **by** 1 **do**

    Compute $\alpha_\ell[p]$ by Eq. (15)-(17) or Eq. (18) ; // Compute attribution score

    $B_\ell[p, q] \leftarrow \alpha_\ell[p, q] \times \beta_\ell[p], \quad \forall\ q \in [d_{\ell-1}]$; // Attribute to the layer input

  end

  for $q \leftarrow 1$ **to** $d_{\ell-1}$ **by** 1 **do**

    $\beta_{\ell-1}[q] \leftarrow \prod_{i=\ell}^m W_i^x b_{j-1}^x + \sum_{p=1}^{d_\ell} B_\ell[p, q]$; // Combine with bias of layer $\ell - 1$

  end

end

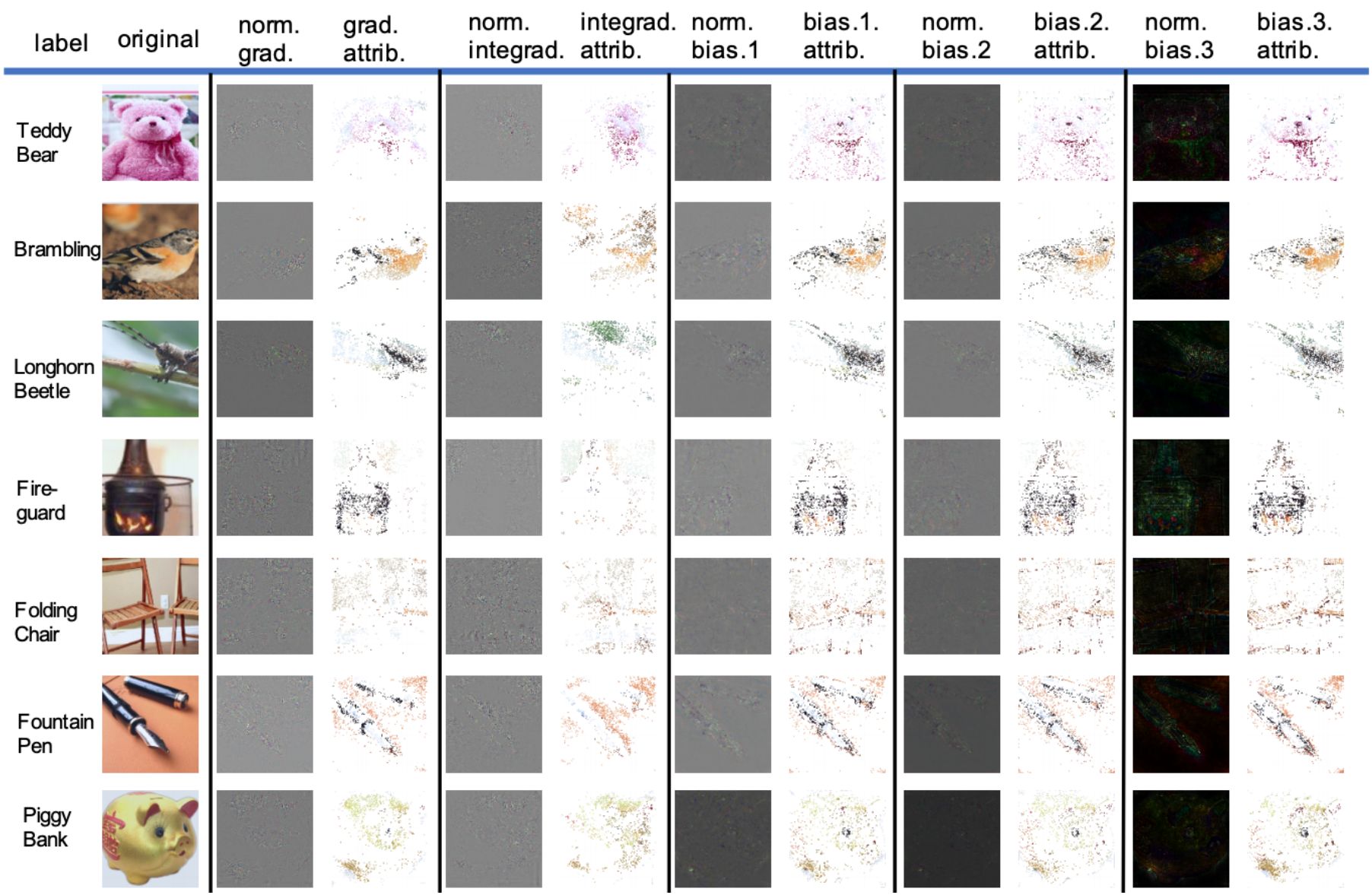
return $\beta_1 \in \mathbb{R}^{d_{in}}$

Experiments

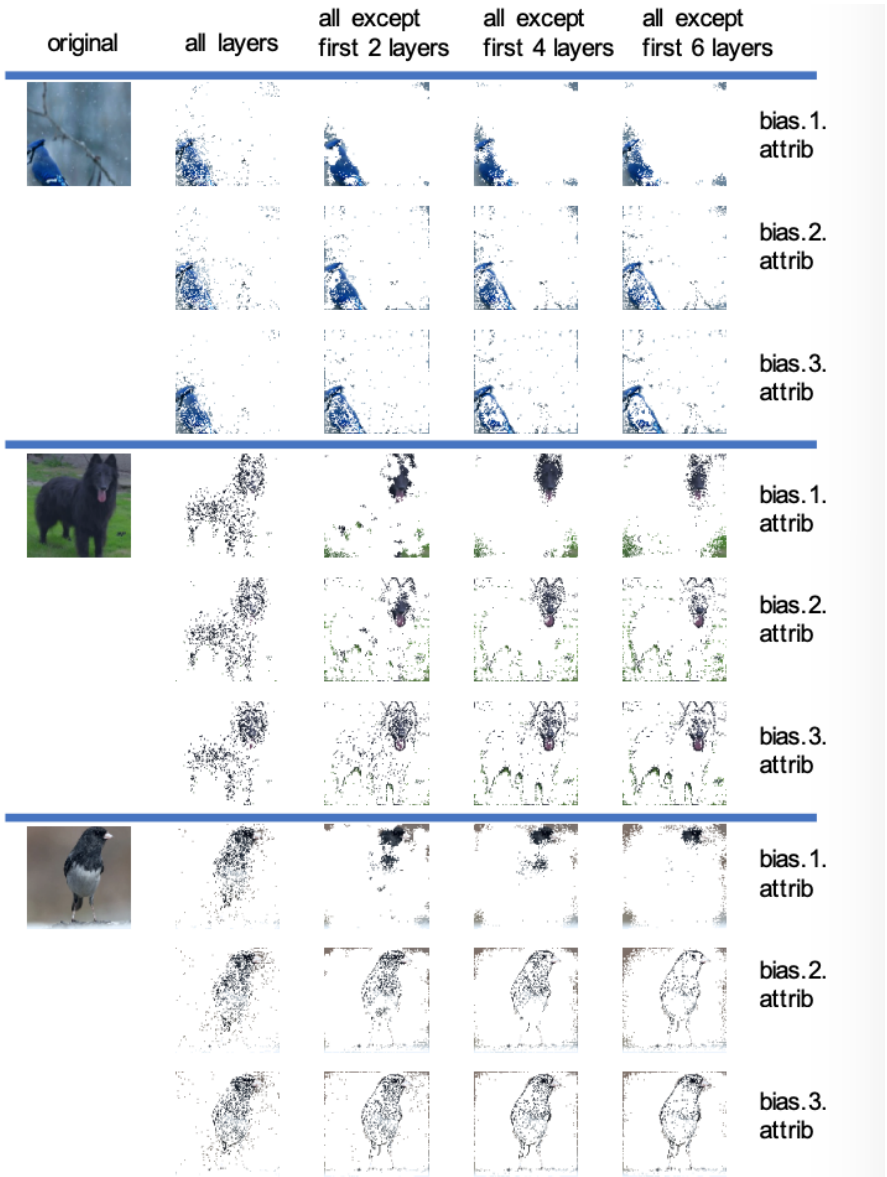
- Importance of Bias in DNNs
- Bias Attribution Analysis Visualization
- Bias Attribution for Various Layers

Dataset	Train Without Bias	Train With Bias, Test All	Test Only wx	Test Only b
CIFAR10	87.0	90.9	71.5	62.2
CIFAR100	62.8	66.8	40.3	36.5
FMNIST	94.1	94.7	76.1	24.6

Bias Attribution Analysis Visualization



Bias Attribution for Various Layers



Conclusions

- the bias in a DNN also has a non-negligible contribution to the correctness of predictions, it can also play a significant role in understanding DNN behavior.
- a backpropagation-type algorithm "bias back-propagation (BBp)" that starts at the output layer and iteratively attributes the bias of each layer to its input nodes as well as combining the resulting bias term of the previous layer.