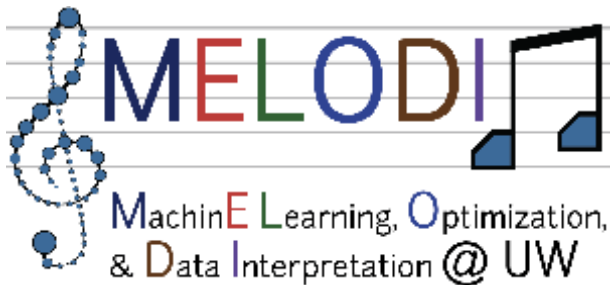


Bias Also Matters: Bias Attribution for Deep Neural Network Explanation

Shengjie Wang*, Tianyi Zhou*, Jeff A. Bilmes

University of Washington, Seattle



Explain DNNs as a linear model per data point

- DNN with piecewise linear activations like ReLU, when applied to a data point x , equals to a linear model $g(x) = wx + b$.
- The gradient term, i.e., w in $g(x)$, has been widely studied to explain DNN output on a given data point.
- The bias b , however, is usually overlooked.

Bias contains important information of DNNs

- Decomposition of a DNN for every data point x :

$$f(x) = W_m \psi_{m-1}(W_{m-1} \psi_{m-2}(\dots \psi_1(W_1 x + b_1) \dots) + b_{m-1}) + b_m$$

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

- The bias term, though as a scalar, results from the complicated process involving both the weights and biases of DNN layers.

Bias is important for DNN performance

- Linear model with gradient term only may produce wrong predictions.
- The bias term corrects it.

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

Our method “Bias Backpropagation (BBp)” explicitly attributes the bias term to each input feature.




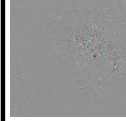

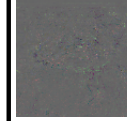

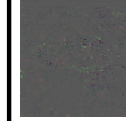

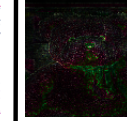


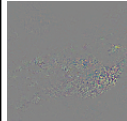

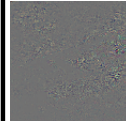
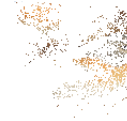
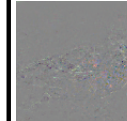

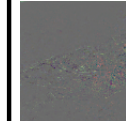

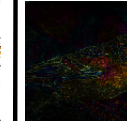

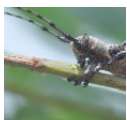
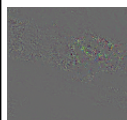
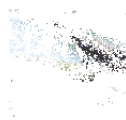
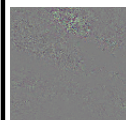

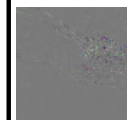
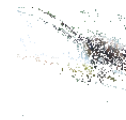
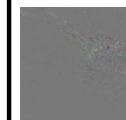

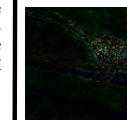










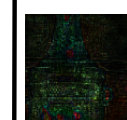




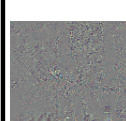




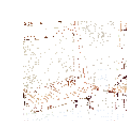
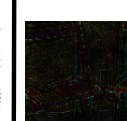

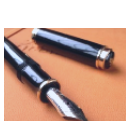


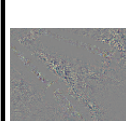

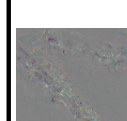

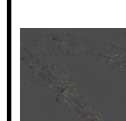

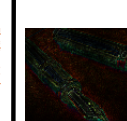


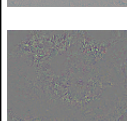

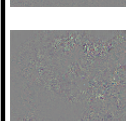



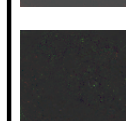
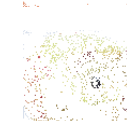
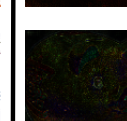

Bias Backpropagation (BBp)

- Start from the final layer and attribute the bias in a backpropagation style.
- For every layer:
 - Receive the bias attribution from the previous layer.
 - Combine the received bias attribution with the effective bias of this layer.
 - Attribute the combined term to the input of this layer.
- The sum of attribution on all input features exactly recovers b^x .

Algorithm 1 Bias Backpropagation (BBp)

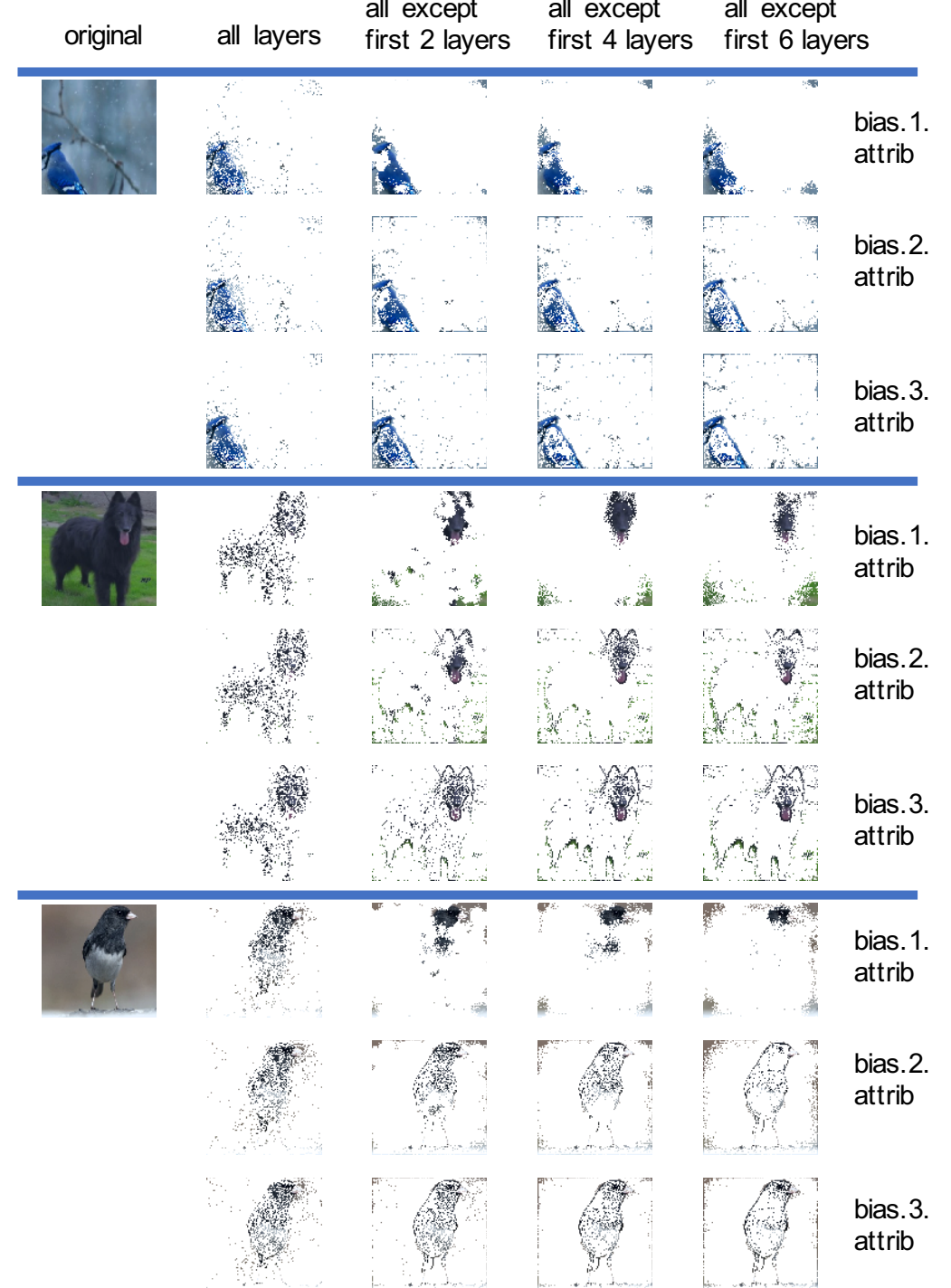
```
input :  $x, \{W_\ell\}_{\ell=1}^m, \{b_\ell\}_{\ell=1}^m, \{\psi_\ell(\cdot)\}_{\ell=1}^m$ 
1 Compute  $\{W_\ell^x\}_{\ell=1}^m$  and  $\{b_\ell^x\}_{\ell=1}^m$  for  $x$  by Eq. (5); // Get
   data point specific weight/bias
2  $\beta_m \leftarrow b_m$ ; //  $\beta_\ell$  holds the accumulated attribution for
   layer  $\ell$ 
3 for  $\ell \leftarrow m$  to 2 by -1 do
4   for  $p \leftarrow 1$  to  $d_\ell$  by 1 do
5     Compute  $\alpha_\ell[p]$  by Eq. (15)-(17) or Eq. (18);
     // Compute attribution score
6      $B_\ell[p, q] \leftarrow \alpha_\ell[p, q] \times \beta_\ell[p], \forall q \in [d_{\ell-1}]$ ;
     // Attribute to the layer input
7   end
8   for  $q \leftarrow 1$  to  $d_{\ell-1}$  by 1 do
9      $\beta_{\ell-1}[q] \leftarrow \prod_{i=\ell}^m W_i^x b_{\ell-1}^x + \sum_{p=1}^{d_\ell} B_\ell[p, q]$ ;
     // Combine with bias of layer  $\ell - 1$ 
10  end
11 end
12 return  $\beta_1 \in \mathbb{R}^{d_{in}}$ 
```

Examples of Attribution Results on Images

label	original	norm. grad.	grad. attrib.	nom. integrad.	integrad. attrib.	nom. bias.1	bias.1. attrib.	nom. bias.2	bias.2. attrib.	norm. bias.3	bias.3. attrib.
Teddy Bear											
Brambling											
Longhorn Beetle											
Fire-guard											
Folding Chair											
Fountain Pen											
Piggy Bank											

Bias Attribution of various layers

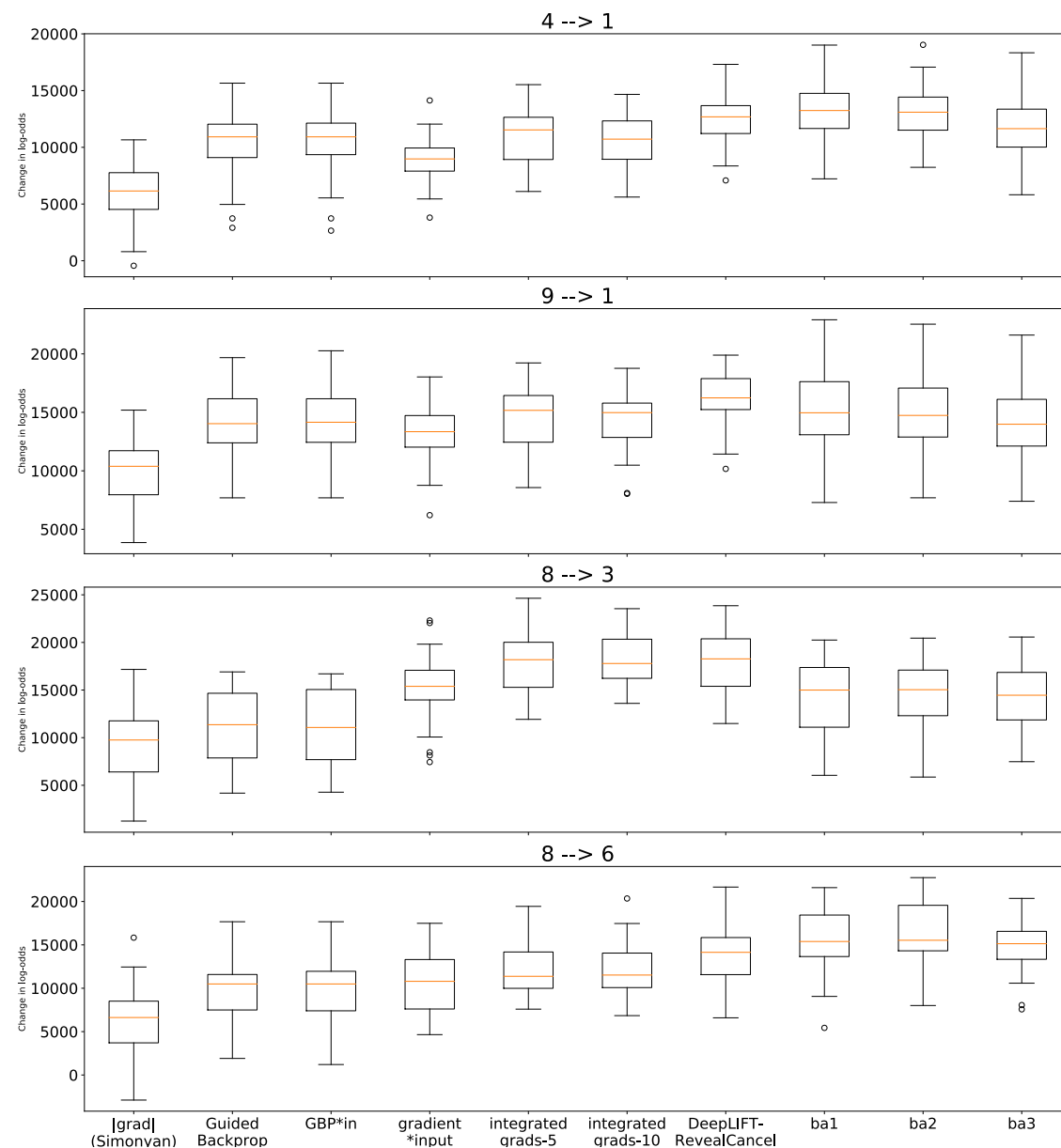
- We can use BBp to analyze biases of different layers.
- Bias from lower layers results in more noise in the attribution.
- Bias from deeper layer reveals high-level features (e.g., head parts of the dog and the bird).



“bias.1(2,3)” corresponds to the three variants of BBp.

Quantitative evaluation on MNIST digit flip test

- Mask input image pixels based on the attribution scores.
- Check the change of the predictions.
- Log-odds scores of target vs. source class before and after masking pixels.
- BBp is class-sensitive and comparable to methods such as integrated gradient and DeepLift.



Thank you!

- For more details, please come to our poster session

Wednesday 06:30 - 09:00 PM
Pacific Ballroom #147