NIF: A Framework for Quantifying Neural Information Flow in Deep Networks

Brian Davis*, Umang Bhatt*, Kartikeya Bhardwaj*, Radu Marculescu, José Moura
Carnegie Mellon University, Pittsburgh, PA, USA
{briandavis, umang, kbhardwa, radum, moura}@cmu.edu

Overview

- With the rise of deep learning, network interpretability of deep networks has emerged as a challenging problem. An information-theoretic understanding of deep networks is particularly lacking.
- Current methods of estimating mutual information do not consider the flow between individual neurons.
- We propose a method utilizing MINE [1] to estimate the mutual information between neurons in a network. We accomplish this by removing the redundant information within a layer from the information calculated between a layer and an individual neuron.
- We also explore how this technique can be utilized to create feature attributions to provide better insight into how the model prioritizes input features.

Information Measures

- Mutual Information (MI) is defined as: $I(X,Y) = H(X) - H(X|Y)$ and is the reduction of uncertainty in X given Y. We wish to estimate $I(\mathcal{X}_i; \mathcal{Q}_k)$, the MI between 2 nodes in a trained network.
- Since calculation of this quantity is intractable, we exploit the MINE [1] estimator which uses a statistics network T_θ to approximate the following:

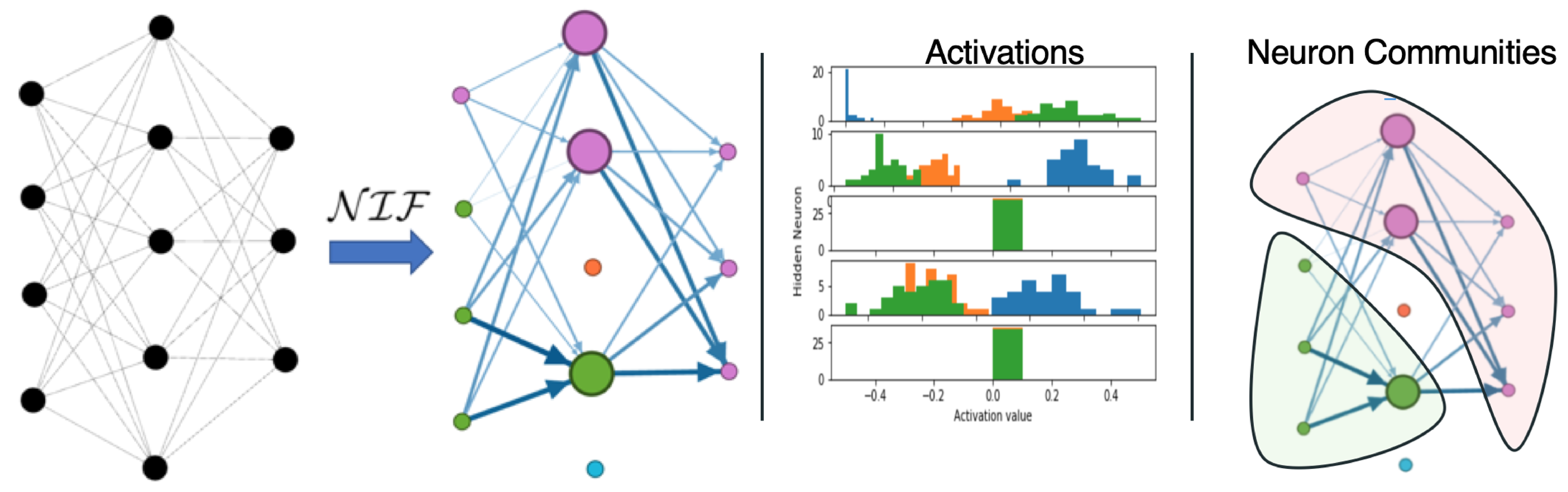
$$\widehat{I}(\mathcal{X}, \mathcal{Z}) = \sup_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}_{\mathcal{X}\mathcal{Z}}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_{\mathcal{X}} \otimes \mathbb{P}_{\mathcal{Z}}}[e^{T_\theta}]) \quad (1)$$

- We decompose this approximation to give us $I(\mathcal{X}_i; \mathcal{Q}_k)$, where X_i is a feature of the input vector and Q_k is any quantity of interest. That is, we leverage an approximation [2]:

$$I(\mathcal{X}_i; \mathcal{Q}_k) = I(\mathcal{X}; \mathcal{Q}_k) - \beta \sum_{j=1}^{i-1} I(\mathcal{X}_i; \mathcal{X}_j) \quad (2)$$

  where β can be used to tune the interactive effect of MI between features.
- The first term is referred to as the *relevance* of $\mathcal{X}$ to $\mathcal{Q}_k$ and the second term is called *redundancy*, as it removes interactions between dimensions of the input.

Approach



Since T_θ shares model parameters between the *redundancy* (A) and *relevance* (B) components, we derive a weaker least upper bound. To better understand distributional interactions, we define the following:

$$A = \mathbb{E}_{\mathbb{P}_{\mathcal{X}\mathcal{Q}_k}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_{\mathcal{X}} \otimes \mathbb{P}_{\mathcal{Q}_k}}[e^{T_\theta}])$$

$$B = \mathbb{E}_{\mathbb{P}_{\mathcal{X}_i \mathcal{X}_j}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_{\mathcal{X}_i} \otimes \mathbb{P}_{\mathcal{X}_j}}[e^{T_\theta}])$$

We combine these parameters to derive NIF:

$$\mathcal{NIF} = \sup_{\theta \in \Theta} \big(A - \beta \sum_{j=1}^{i-1} B\big) \geq \widehat{I}(\mathcal{X}_i, \mathcal{Q}_k, T_\theta) \quad (3)$$

Feature Attribution

- To recover a feature attribution, we find all the possible paths between a feature of interest and each of the outputs.
- Mathematically, the element $\mathcal{A}_{i,j}$ of our *attribution matrix* $\mathcal{A} \in \mathbb{R}^{d \times c}$ (where d is the number of features and c is the number of classes) can be given as:

$$\mathcal{A}_{ij} = \sum_{C_j} \sum_{p_{ij} \in \mathbb{P}} \prod_{\ell \in p_{ij}} I(\ell_{\text{start}}, \ell_{\text{end}}) \quad (4)$$

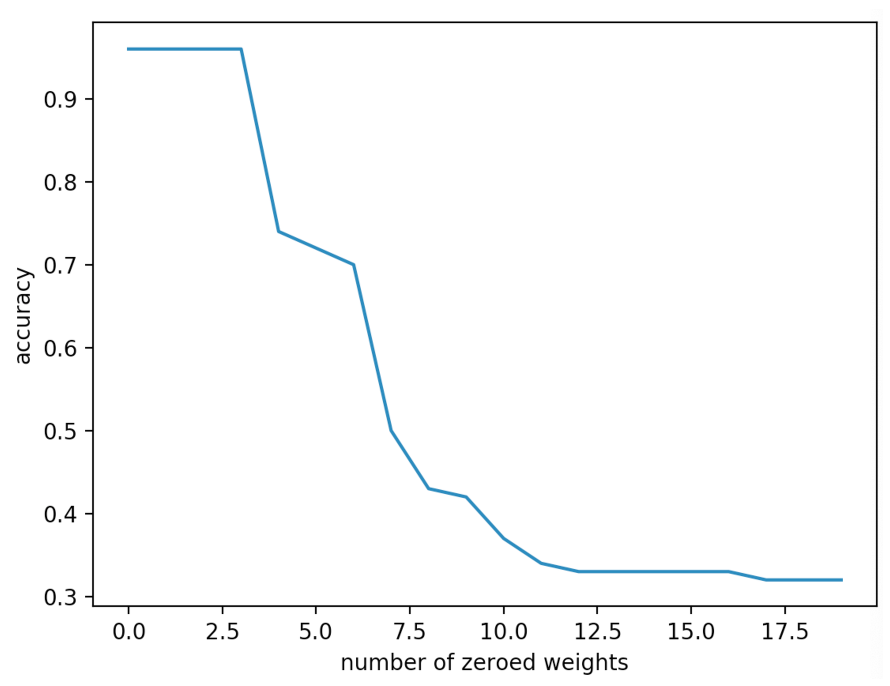
  where, $\mathbb{P}$ is the set of all directed paths from input x_i to class y_j in the NIF network, and $\mathbb{L}$ is the set of links on each path $p \in \mathbb{P}$.

Experimentation

To prove the fidelity of NIF, we run experiments to extract a feature attribution to explain the original model output and to prune the network for compression.
We conducted our experiments on the Iris and Banknote dataset.

Implications for Model Compression

- Often in neural network training, many neurons learn information which is not necessary for final prediction
- Such useless neurons can be removed as they only lead to unnecessary computation without affecting model accuracy
- NIF naturally enables detection of such neurons from an information-theoretic standpoint – We identify neurons that have zero information flowing through them
- Zeroing out weights and biases of these neurons does not affect classification accuracy.



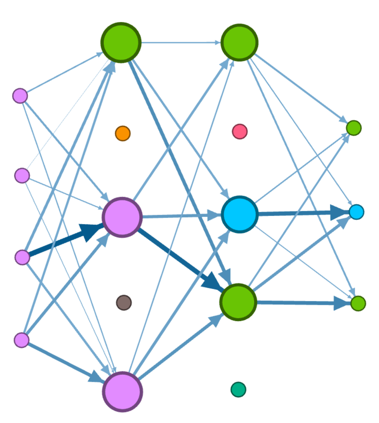
Feature Attribution

- We evaluate the feature attribution provided by NIF against current techniques.
- Via the K-S test, we observe that the raw mutual information and the NIF attribution are likely drawn from the same distribution.

Attribution	K-S statistic	p-value
NIF	1.0	0.011
SHAP[3]	0.75	0.107
IG[4]	0.25	0.996

Multilayer Networks

- We conducted further experiments on deeper neural architectures.
- We observed similar behaviors in communities and zero information neurons.



Conclusion

- We have proposed NIF, Neural Information Flow, a new metric for measuring information flow through deep learning models.
- Merging a dual representation of Kullback-Leibler divergence and classical feature selection literature, we find that NIF provides insight into which information pathways are crucial within a network.
- We show that the feature importance captured by NIF rivals prior techniques from an information-theoretic perspective.
- NIF can also allow us to leverage fewer parameters at inference time, since we can remove parameters deemed useless by the NIF without loss of accuracy.

References

[1] Belghazi et al. In *Proceedings of the 35th ICML*, 2018.

[2] K. D. Bollacker and J. Ghosh. Linear feature extractors based on mutual information. In *Proceedings of 13th ICPR*, 1996.

[3] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NIPS 30*. 2017.

[4] Marco Ancona et al. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*, 2018.