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, Pennsylvania 15213-3890
{briandavis, umang, kbhardwa, radum, moura}@cmu.edu

Abstract

In this paper, we present a new approach to interpreting deep learning models. More precisely, by coupling mutual information with network science, we explore how information flows through feed forward networks. We show that efficiently approximating mutual information via the dual representation of Kullback-Leibler divergence allows us to create an information measure that quantifies how much information flows between any two neurons of a deep learning model. To that end, we propose NIF, Neural Information Flow, a new metric for codifying information flow which exposes the internals of a deep learning model while providing feature attributions.

Introduction

As deep learning gains popularity, there has been an influx in methods that attempt to explain how deep learning begets its predictive power. Most approaches to interpreting deep learning models are model agnostic and make local approximations in the feature space region around the datapoints to be explained (Ribeiro, Singh, and Guestrin 2016). However, such techniques fail to capture global *model-specific* behavior that is crucial to understanding if the function learned by a deep learning model aligns well with a users' intention. Moreover, current noisy approximations neglect the topological structure of model used for prediction (Sundararajan, Taly, and Yan 2017).

We note that it is easy to forget the network structure of deep learning models, particularly feed forward models, which resemble directed acyclic graphs. However, understanding the topological structure of different models can not only help decide the architecture best suited for the task at hand, but also help expose the internal interactions between neurons at inference time. While the existing interpretability techniques (Chen et al. 2018) shed light on which input features are responsible for a given prediction, prior art still fails to quantify how information flows through a deep network at the neuron-level. This prevents answering one of the most fundamental questions in deep learning: *How much information flows through a deep network model from input features to each of its intermediate neurons?*

To address this question we consider two types of interpretability notions: (*i*) model interpretability via attribution to input features, and (*ii*) network architecture interpretability with respect to how information flows from neuron to neuron within a given pretrained model. We believe, addressing notion (*ii*) above from the fundamental information theory standpoint will automatically reveal insights about the precise decision-making process followed by the model (i.e., the notion (*i*) above).

To that end, using an information theoretic measure, we propose to model the flow of information via *Neural Information Flow* (NIF) between neurons in consecutive layers to expose how simple deep learning models can learn complex functions of their input features. We further analyze this flow of information between neurons from a network science (Barabási and Bonabeau 2003) perspective, where each neuron in the deep network essentially becomes a node in the network of information flow. Eventually, the NIF can help recover an information-theoretic feature attribution, a rank of feature importance to a given class.

Combining an information measure with the ability to propagate information through the network can help us visualize the information flow. Feature attributions not only expose which features are important to a model (just like current feature attribution techniques (Ribeiro, Singh, and Guestrin 2016)), but also which information flow paths in the network are crucial to a model's prediction; the latter will allow us to study how information flow is amplified or thwarted when we introduce current state of the art deep learning building blocks: shortcuts, residuals, dropout, etc. To the best of our knowledge, we are the first to propose an information- and network-theoretic model for explaining how information flows through a deep learning model while accounting for its network structure.

Background

Network Science

Network science has gained a lot of interest for many biological and social science applications. However, to the best of our knowledge, network concepts have *not* been used to understand the inner workings of deep neural networks. To that end, several ideas from network science can be used for better understanding deep network architectures.

*Equal Contribution

Betweenness Centrality Given a network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, betweenness centrality $B(v)$ of a node $v \in \mathcal{V}$ is a measure of how central a node is in the network. Specifically, the $B(v)$ computes how many shortest paths between different pairs of nodes in the network pass through node v. Mathematically, betweeness can be computed as:

$$B(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where, σ_{st} is the number of shortest paths between nodes $s, t \in \mathcal{V}$, and $\sigma_{st}(v)$ are the shortest paths passing through v.

Community Structure Communities in a network refer to groups of tightly connected nodes. Intuitively, a community can be defined as a group of nodes if the number of connections between this group is significantly more than what we would expect at random. Mathematically, communities can be computed by maximizing a modularity function as follows (Newman 2006):

$$\max_{\mathbf{g}=\{g_1,g_2,\ldots,g_k\}} \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{1}{\gamma} \cdot \frac{k_i k_j}{2m} \right] \delta(g_i, g_j) \quad (1)$$

where, m is the number of edges, k_i is the degree (number of connections) of node i, A_{ij} is the weight of the link between nodes i and j, and δ is Kronecker delta. The idea is to find groups of tightly connected nodes, $\mathbf{g} = \{g_1, g_2, \ldots, g_k\}$, which map the nodes $\mathcal{V}$ to k communities. The $k_i k_j / 2m$ factor represents the number of links one would expect in a randomly connected network. Finally, γ controls the resolution of communities: lower gamma will detect more number of smaller communities (i.e., the number of communities k depends on the resolution of communities, γ).

Interpretability

Current interpretability techniques fall into two classes. The first class of work are gradient-based methods, which compute the gradient of the output with respect to the input, treating gradient flow as a saliency map (Sundararajan, Taly, and Yan 2017). The other type of research leverages perturbation-based techniques to approximate a complex model using a locally additive model, thus explaining the difference between test output-input pair and some reference output-input pair. Lundberg and Lee proposed SHAP, a class of methods which randomly draws points from a kernel centered at the test point and fits a sparse linear model to locally approximate the decision boundary (Lundberg and Lee 2017). Approximating Shapley values to quantify the importance of features of a given input, kernel SHAP can learn a feature attribution. While gradient-based techniques like (Sundararajan, Taly, and Yan 2017) consider infinitesimal regions on the decision surface and take the first-order term in the Taylor expansion as the additive model, perturbation-based additive models consider the finite difference between an input vector and a reference vector.

Information Theory

Mutual information has proven to be a valuable tool for feature selection at training time leveraging dimensionality reduction (Bollacker and Ghosh 1996). More recent work has represented deep neural networks as Markovian chains to create an information bottleneck theory for deep learning (Shwartz-Ziv and Tishby 2017). However, these works do not tackle the interpretability problem directly.

Other works look to find $I(\mathcal{X}; \mathcal{Y})$, the mutual information between a subset of the input vector and the output vector. In order to explain the conditional distribution of the output vector given the input vector, Chen et al. develop an efficient variational approximation to mutual information (Chen et al. 2018). However, this model fails to recover the per-feature mutual information, a requisite of our model to explain how information flows through all possible paths.

Additionally, (Belghazi et al. 2018) proposes to estimate mutual information via a *neural information measure*, $I_\Theta(\mathcal{X}, \mathcal{Z})$: this quantity is grounded in the dual representation of the Kullback-Leibler divergence between the joint and product of the marginals parameterized by $\theta \in \Theta$ from a *statistics network* $T_\Theta : \mathcal{X} \times \mathcal{Z} \to R$: a deep neural network used to estimate the neural information measure from empirical samples from the joint ($P_{\mathcal{X}\mathcal{Z}}$) and the product of the marginal distributions ($P_{\mathcal{X}} \otimes P_{\mathcal{Z}}$). The empirical neural information measure ($\widehat{I}$) is defined as follows:

$$\widehat{I}(\mathcal{X}, \mathcal{Z}) = \sup_{\theta \in \Theta} E_{P_{\mathcal{X}\mathcal{Z}}}[T_\theta] - \log(E_{P_{\mathcal{X}} \otimes P_{\mathcal{Z}}}[e^{T_\theta}])$$

We use this approximation to start our detailed understanding of information flow in deep networks. We control $\mathcal{X}$ and $\mathcal{Z}$ to be different quantities of interest within our model, $\mathcal{M}$, namely a specific input feature, a hidden neuron, etc.

Approach

Our proposed approach, NIF, transforms a traditional deep learning model into a representation that actually captures the information-theoretic relationship between nodes learned by the model (Fig 1).

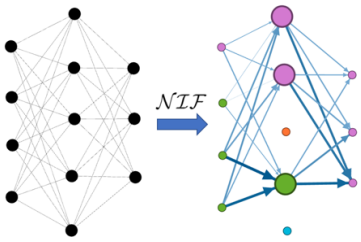


Figure 1: Traditional Model to NIF Network. Color of nodes corresponds to communities. Size of nodes corresponds to betweenness centrality

Our approach extends the work of (Belghazi et al. 2018) and decomposes their approximation of $I(\mathcal{X}; \mathcal{Z})$ to give us $I(\mathcal{X}_i; \mathcal{Q}_k)$, where X_i is a dimension of X (specifically the i^{th} feature of the input vector) and Q_k is any quantity of interest (perhaps the k^{th} neuron in a hidden layer or a class of the output vector).

Assuming that mutual information is composable and entropy is non-decreasing, we can calculate the mutual infor-

mation for any feature $\mathcal{Q}_k$) by leveraging a tractable approximation from (Bollacker and Ghosh 1996):

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

where β can be used to tune the interactive effect of mutual information 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. We desire a tractable approximation to Equation (2) using the *statistics network*[1], T_Θ, that calculates the mutual information between two empirical distributions (in this case, $\mathcal{X}$ to $\mathcal{Q}_k$). We can find the *relevance* term via:

$$\widehat{I}(\mathcal{X}, \mathcal{Q}_k, T_\Theta) = \sup_{\theta \in \Theta} \Big[E_{P_{\mathcal{X}\mathcal{Q}}}[T_\theta] - \log(E_{P_\mathcal{X} \otimes P_\mathcal{Q}}[e^{T_\theta}]) \Big]$$

Similarly, the *redundancy* term is as follows:

$$\widehat{I}(\mathcal{X}_i, \mathcal{X}_j, T_\Theta) = \sup_{\theta \in \Theta} \Big[E_{P_{\mathcal{X}_i\mathcal{X}_j}}[T_\theta] - \log(E_{P_{\mathcal{X}_i} \otimes P_{\mathcal{X}_j}}[e^{T_\theta}]) \Big]$$

Combining both *relevance* and *redundancy*, we get the following estimate of neural information:

$$\widehat{I}(\mathcal{X}_i, \mathcal{Q}_k, T_\Theta) = \widehat{I}(\mathcal{X}, \mathcal{Q}_k, T_\Theta) - \beta \sum_{j=1}^{i-1} \widehat{I}(\mathcal{X}_i, \mathcal{X}_j, T_\Theta)$$

Since we share model parameters between the *redundancy* and *relevance* components, we derive a weaker least upper bound that allows us to get granular about distributional interactions. First, let the following hold:

$$A = E_{P_{\mathcal{X}\mathcal{Q}}}[T_\theta] - \log(E_{P_\mathcal{X} \otimes P_\mathcal{Q}}[e^{T_\theta}])$$

$$B = E_{P_{\mathcal{X}_i\mathcal{X}_j}}[T_\theta] - \log(E_{P_{\mathcal{X}_i} \otimes P_{\mathcal{X}_j}}[e^{T_\theta}])$$

To that end, we propose NIF, a new metric for neural information flow.

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

By jointly training T_Θ, we can approximate the mutual information between a feature and a quantity of interest; for concreteness, let's assume the quantity of interest is the first hidden neuron of a hidden layer. Solving Equation (3) for all possible $\mathcal{X}_i$ will place a weight on every edge between a feature and a quantity of interest.[2]

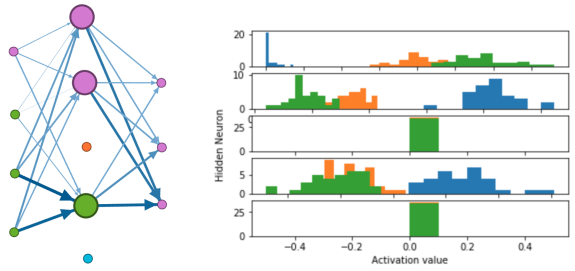
Results

In order to test the fidelity of NIF, we run a few experiments that not only validate our proposed metric, but also lead to novel interpretations of deep learning models. We run all experiments on UCI datasets, namely Iris and Banknote authentication, both of which provide us with a small enough feature space for us to interpret and visualize (Dheeru and Karra Taniskidou 2017).

[1]For a thorough derivation of the statistics network as a valid measure of mutual information, see (Belghazi et al. 2018).

[2]Note we can scale $\mathcal{X}_i$ and $\mathcal{Q}_k$ to be any two model internals.

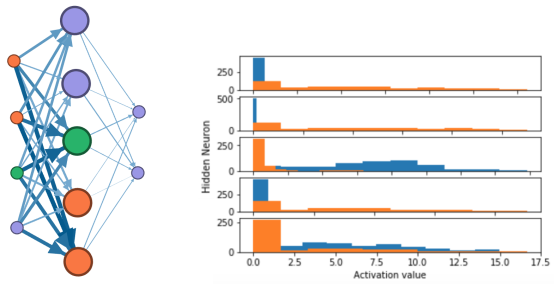
One Layer Perceptron We start by visualizing NIF for a one layer perceptron trained on the Iris dataset with ReLU activations and optimized via ADAM. Note that we make a feature independence assumption for the Iris dataset, since its low number of samples hinders NIF convergence.



(a) (b)

Figure 2: One layer perceptron for the Iris dataset with ReLU activation and trained with ADAM (a) NIF network model (b) Activation distribution of the original model

In Figure 2, we show the NIF network created using Equation (3) and a distribution of activations, as a sanity check. Particularly, in Figure 2(a), we normalize the information flow per layer to ease visualization of the edges. The thickness of an edge denotes how much information is flowing between any two nodes: the thicker the connection, the more information travelling form one node to the next. The size of the node denotes its centrality: the bigger the node, the more central it is for information to freely propagate through the network. The color of the node denotes which community the node is a member of: using a standard resolution of $\gamma = 1$, we use Equation (1) to find three distinct communities in the network. Upon first glance, it is clear that of the five hidden neurons in the one hidden layer, only three are central to the model's final prediction. This results makes intuitive sense as ReLU activation at those nodes is zero (see Figure 2(b)): thus, we can reason that ReLU effectively stifles information from flowing through the network. Moreover, Figure 2(b) confirms that the distribution of activations at nodes three and five are zero and, therefore, have no connections in the NIF model.



(a) (b)

Figure 3: One layer perceptron for the Banknote dataset with ReLU activation and trained with ADAM (a) NIF network (b) Activation distribution of the original model

We perform similar analysis for the Banknote dataset and report results in Figure 3. We see a strong information propagation from feature one to hidden layer node five, so much so that both nodes belong to their own community. Leveraging the activation distribution in Figure 3(b) confirms the equal importance of all central nodes to the model's prediction.

Two Layered Network To show the initial ability of NIF to generalize larger networks, we train a two layered network with ReLU activation on the Banknote dataset. Shown in Figure 4, we find that two nodes per layer are zero which means there are information pathways that are inherently stifled due to use of ReLU activation.

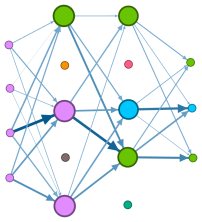


Figure 4: NIF network for a two layer MLP for the Banknote dataset with ReLU activation and trained with ADAM

Accuracy Recovery It is worthwhile to note that all of the models described above received upwards of 96% accuracy on a held-out test set. The NIF model shown in Figure 2 and in Figure 4 found that ReLU can zero out activations at certain neurons in the hidden layer of a network while still passing enough information through the rest of the neurons to maintain predictive accuracy. We ran another set of experiments wherein we zero out the weights and biases of the original model for zero activation neurons in the original model. We find that using NIF to identify useless weights (and then acting upon that learning to zero out the corresponding elements in the weight matrix). To our surprise, we did **not** find a drop in accuracy when we zeroed out the weights in the matrix. This will have massive implications as we scale to larger networks.

Feature Attribution NIF naturally recovers a feature attribution which we calculate in the following manner. We find all the possible paths between a feature of interest x_i and any of the outputs $y_1, \ldots, y_c$. To find out the value of a path, we take the product of all NIF calculations along the path. We then sum over all of the possible values to find $\mathcal{A}_{i,j}$, our desired feature attribution for feature x_i and class y_j. Mathematically, the element $\mathcal{A}_{i,j}$ of our *attribution matrix* $\mathcal{A} \in R^{n \times c}$ (where, n is the number of features and c is the number of classes) can be given as:

$$\mathcal{A}_{ij} = \sum_{p \in P} \prod_{l \in L} \mathcal{NIF}_p(l)$$

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

We compare NIF to current feature attribution techniques SHAP (Lundberg and Lee 2017) and Integrated Gradients (Sundararajan, Taly, and Yan 2017) in Table 1. Using the two sample Kolmogorov-Smirnov test for goodness of fit between two empirical distributions (in this case, the raw mutual information attribution between the input and output classes and the attribution in question), we find that NIF surpasses current benchmarks, which means NIF is likely drawn from the same distribution as the raw mutual information between the input and output classes. This leads us to believe that information theoretic feature attribution is viable.

ATTRIBUTION	K-S STATISTIC	P-VALUE
NIF	1.0	0.011
SHAP	0.75	0.107
INTEGRATED GRADIENTS	0.25	0.996

Table 1: Feature attribution comparison

Conclusion and Future Work

We have proposed NIF, Neural Information Flow, a new metric for measuring information flow through deep learning models. Merging a dual representation of Kullback-Leilber divergence and classical feature selection literature, we find that NIF not only provides insight into which information pathways are crucial within a network but also allows us to leverage fewer parameters at inference time, since we can remove parameters deemed useless by the NIF without loss of accuracy. Finally, we have shown how NIF recovers an information theoretic feature attribution that aligns with existing benchmarks. In our future work, we plan to apply NIF to larger architectures.

References

Barabási, A.-L., and Bonabeau, E. 2003. Scale-free networks. *Scientific American* 288(60-69).

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D. 2018. Mutual information neural estimation. In *Proc. ICML*, volume 80, 531–540. PMLR.

Bollacker, K. D., and Ghosh, J. 1996. Linear feature extractors based on mutual information. In *Proc. ICPR*, 720–724 vol.2.

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2018. Learning to explain: An information-theoretic perspective on model interpretation. *ICML 2018*.

Dheeru, D., and Karra Taniskidou, E. 2017. UCI machine learning repository.

Lundberg, S. M., and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*. 4765–4774.

Newman, M. E. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103(23):8577–8582.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proc. KDD*, 1135–1144.

Shwartz-Ziv, R., and Tishby, N. 2017. Opening the black box of deep neural networks via information. *CoRR*.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *Proc. ICML*, volume 70, 3319–3328. PMLR.