Understanding Impacts of High-Order Loss Approximations and Features in Deep Learning Interpretation

Sahil Singla [1] **Eric Wallace** [1] **Shi Feng** [1] **Soheil Feizi** [1]

Abstract

Current saliency map interpretations for neural networks generally rely on two key assumptions. First, they use first-order approximations of the loss function, neglecting higher-order terms such as the loss curvature. Second, they evaluate each feature's importance in isolation, ignoring feature interdependencies. This work studies the effect of relaxing these two assumptions. First, we characterize a closed-form formula for the input Hessian matrix of a deep ReLU network. Using this, we show that, for classification problems with many classes, if a prediction has high probability then including the Hessian term has a small impact on the interpretation. We prove this result by demonstrating that these conditions cause the Hessian matrix to be approximately rank one and its leading eigenvector to be almost parallel to the gradient of the loss. We empirically validate this theory by interpreting ImageNet classifiers. Second, we incorporate feature interdependencies by calculating the importance of group-features using a sparsity regularization term. We use an $L_0 - L_1$ relaxation technique along with proximal gradient descent to efficiently compute group-feature importance values. Our empirical results show that our method significantly improves deep learning interpretations.

1. Introduction

The growing use of deep learning in sensitive applications such as medicine, autonomous driving, and finance raises concerns about human trust in machine learning systems. For trained models, a central question is test-time *interpretability*: how can humans understand the reasoning behind model predictions? A common interpretation approach is to identify the importance of each input feature for a model's prediction. A saliency map can then visualize the important features, e.g., the pixels of an image (Simonyan et al., 2014; Sundararajan et al., 2017) or words in a sentence (Li et al., 2016).

Several approaches exist to create saliency maps, largely based on model gradients. For example, Simonyan et al. (2014) compute the gradient of the class score with respect to the input, while Smilkov et al. (2017) average the gradient from several noisy versions of the input. Although these gradient-based methods can produce visually pleasing results, they often weakly approximate the underlying model (Feng et al., 2018; Nie et al., 2018). Existing saliency interpretations mainly rely on two key assumptions:

- **Gradient-based loss surrogate:** For computational efficiency, several existing methods, e.g., Simonyan et al. (2014); Smilkov et al. (2017); Sundararajan et al. (2017), assume that the loss function is almost linear at the test sample. Thus, they use variations of the input gradient to compute feature importance.
- **Isolated feature importance:** Current methods evaluate the importance of each feature in isolation, assuming all other features are fixed. Features, however, may have complex interdependencies that can be learned by the model.

This work studies the impact of relaxing these two assumptions in deep learning interpretation. To relax the first assumption, we use the second-order approximation of the loss function by keeping the Hessian term in the Taylor expansion of the loss. For a deep ReLU network and the cross-entropy loss function, we compute this Hessian term in *closed-form*. Using this closed-form formula for the Hessian, we prove the following for ReLU networks:

Theorem 1 (informal version) *If the probability of the predicted class is close to one and the number of classes is large, first-order and second-order interpretations are sufficiently close to each other.*

We present a formal version of this result in Theorem 5 and also validate it empirically. For instance, in ImageNet

[1]Computer Science Department, University of Maryland. Correspondence to: Sahil Singla <ssingla@cs.umd.edu>, Soheil Feizi <sfeizi@cs.umd.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

2012 (Russakovsky et al., 2015), a dataset of 1,000 classes, we show that incorporating the Hessian term in deep learning interpretation has a small impact for most images.

The key idea of the proof follows from the fact that when the number of classes is large and the confidence in the predicted class is high, the Hessian of the loss function is approximately of rank one. In essence, the largest eigenvalue squared is significantly larger than the sum of squared remaining eigenvalues. Moreover, the corresponding eigenvector is approximately parallel to the gradient vector (Theorem 4). This causes first-order and second-order interpretations to perform similarly. We also show in Appendix F.3 that this result holds empirically for a neural network model that is not piecewise linear. Our theoretical results can also be extended to related problems such as adversarial examples, where most methods are based on the first-order loss approximations (Goodfellow et al., 2015; Madry et al., 2018; Moosavi-Dezfooli et al., 2016).

Next, we relax the isolated feature importance assumption. To incorporate feature interdependencies in the interpretation, we define the importance function over subsets of features, referred to as *group-features*. We adjust the subset size on a per-example basis using an unsupervised approach, making the interpretation *context-aware*. Including group-features in the interpretation makes the optimization combinatorial. To circumvent the associated computational issues, we use an $L_0 - L_1$ relaxation as is common in compressive sensing (Candes & Tao, 2005; Donoho, 2006), LASSO regression (Tibshirani, 1996), and other related problems. To solve the relaxed optimization, we employ proximal gradient descent (Parikh & Boyd, 2014). Our empirical results on ImageNet indicate that incorporating group-features removes noise and makes the interpretation more visually coherent with the object of interest. We refer to our interpretation method based on first-order (gradient) information as the CAFO (Context-Aware First Order) interpretation. Similarly, the method based on second-order information is called the CASO (Context-Aware Second Order) interpretation. We provide open-source code.[1]

2. Problem Setup and Notation

Consider a prediction problem from input variables (features) $X \in \mathcal{X} \subset \mathbf{R}^d$ to an output variable $Y \in \mathcal{Y}$. For example, in the image classification problem, $\mathcal{X}$ is the space of images and $\mathcal{Y}$ is the set of labels $\{1, ..., c\}$. We observe m samples from these variables, namely $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)\}$. Let $\mathbb{P}_{X,Y}$ be the observed empirical distribution.[2] The empirical risk minimization (ERM) approach computes the optimal predictor $f_\theta : \mathcal{X} \to \mathcal{Y}$ for a loss function $\ell(\cdot, \cdot)$ using the following optimization:

$$\min_{\theta \in \Theta} \ \mathbb{E}_{\mathbb{P}_{X,Y}} \left[\ell \left(f_\theta(\mathbf{x}), y \right) \right]. \tag{1}$$

Let $\mathcal{S}$ be a subset of $[d] := \{1, 2, ..., d\}$ with cardinality $|\mathcal{S}|$. For a given sample $(\mathbf{x}, y)$, let $\mathbf{x}(\mathcal{S})$ indicate the features of $\mathbf{x}$ in positions $\mathcal{S}$. We refer to $\mathbf{x}(\mathcal{S})$ as a group-feature of $\mathbf{x}$. The importance of a group-feature $\mathbf{x}(\mathcal{S})$ is proportional to the change in the loss function when $\mathbf{x}(\mathcal{S})$ is perturbed. We select the group-feature with maximum importance and visualize that subset in a saliency map.

Definition 1 (Group-Feature Importance Function) *Let θ^* be the optimizer of the ERM problem (1). For a given sample $(\mathbf{x}, y)$, we define the group-feature importance function $I^{k,\rho}_{\theta^*}(\mathbf{x}, y)$ as follows:*

$$\begin{aligned} I^{k,\rho}_{\theta^*}(\mathbf{x}, y) := \max_{\tilde{\mathbf{x}}} \ & \ell \left(f_{\theta^*}(\tilde{\mathbf{x}}), y \right) \\ & \|\tilde{\mathbf{x}} - \mathbf{x}\|_0 \leq k, \\ & \|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \leq \rho, \end{aligned} \tag{2}$$

where $\|.\|_0$ counts the number of non-zero elements of its argument (known as the L_0 norm). The parameter k characterizes an upper bound on the cardinality of the group-features. The parameter ρ characterizes an upper bound on the L_2 norm of feature perturbations.

If $\tilde{\mathbf{x}}^*$ is the solution of optimization (2), then the vector $|\tilde{\mathbf{x}}^* - \mathbf{x}|$ contains the feature importance values that are visualized in the saliency map. Note, when $k = 1$ this definition simplifies to current feature importance formulations which consider features in isolation. When $k > 1$, our formulation can capture feature interdependencies. Parameters k and ρ in general depend on the test sample $\mathbf{x}$ (i.e., the size of the group-features are different for each image and model). We introduce an unsupervised metric to determine these parameters in Section 4.1, but assume these parameters are given for the time being.

The cardinality constraint $\|\tilde{\mathbf{x}} - \mathbf{x}\|_0 \leq k$ (i.e. the constraint on the group-feature size) leads to a combinatorial optimization problem in general. Such a sparsity constraint has appeared in different problems such as compressive sensing (Candes & Tao, 2005; Donoho, 2006) and LASSO regression (Tibshirani, 1996). Under certain conditions, we show that without loss of generality the L_0 norm can be relaxed with the (convex) L_1 norm (Appendix E).

Our goal is to solve optimization (2) which is non-linear and non-concave in $\tilde{\mathbf{x}}$. Current approaches do not consider the cardinality constraint and optimize $\tilde{\mathbf{x}}$ by linearizing the objective function (i.e., using the gradient). To incorporate group-features into current methods, we can add the constraints of optimization (2) to the objective function using

[1] https://github.com/singlasahil14/CASO

[2] Note that for simplicity, we hide the dependency of $\mathbb{P}_{X,Y}$ on m.

Lagrange multipliers. This yields the following Context-Aware First-Order (CAFO) interpretation function.

Definition 2 (The CAFO Interpretation) *For a given sample* $(\mathbf{x}, y)$*, we define the Context-Aware First-Order (CAFO) importance function* $\tilde{I}^{\lambda_1,\lambda_2}_{\theta^*}(\mathbf{x}, y)$ *as follows:*

$$\tilde{I}^{\lambda_1,\lambda_2}_{\theta^*}(\mathbf{x}, y) := \max_{\Delta} \; \nabla_{\mathbf{x}} \ell\left(f_{\theta^*}(\mathbf{x}), y\right)^t \Delta - \lambda_1 \|\Delta\|_1 - \lambda_2 \|\Delta\|_2^2 \tag{3}$$

where λ_1 *and* λ_2 *are non-negative regularization parameters. We refer to the objective of this optimization as* $\tilde{\ell}(\Delta)$*, hiding its dependency on* $(\mathbf{x}, y)$ *and* θ^* *to simplify notation.*

Large values of regularization parameters λ_1 and λ_2 in optimization (3) correspond to small values of parameters k and ρ in optimization (2). Incorporating group-features naturally leads to a sparsity regularizer through the L_1 penalty. Note, this is not a hard constraint which forces a sparse interpretation. Instead, given proper choice of the regularization coefficients, the interpretation will reflect the sparsity used by the underlying model. In Section 4.1, we detail our method for setting λ_1 on an example-specific basis (i.e., context-aware) based on the sparsity ratio of CAFO's optimal solution. Moreover, in Appendix E, we show that under some general conditions, optimization (3) can be solved efficiently and its solution matches that of the original optimization (2).

To better approximate the loss function, we use its second-order Taylor expansion around point $(\mathbf{x}, y)$:

$$\ell\left(f_{\theta^*}(\tilde{\mathbf{x}}), y\right) \approx \ell\left(f_{\theta^*}(\mathbf{x}), y\right) + \underbrace{\nabla_{\mathbf{x}} \ell\left(f_{\theta^*}(\mathbf{x}), y\right)^t \Delta}_{\text{the first-order term}} + \underbrace{\frac{1}{2} \Delta^t \mathbf{H}_{\mathbf{x}} \Delta}_{\text{the second-order term}} \tag{4}$$

where $\Delta := \tilde{\mathbf{x}} - \mathbf{x}$ and $\mathbf{H}_{\mathbf{x}}$ is the Hessian of the loss function on the input features $\mathbf{x}$ (note y is fixed). This second-order expansion of the loss function decreases the interpretation's model approximation error.

By choosing proper values for regularization parameters, the resulting optimization using the second-order surrogate loss is strictly a convex minimization (or equivalently concave maximization) problem, allowing for efficient optimization using gradient descent (Theorem 3). Moreover, even though the Hessian matrix $\mathbf{H}_{\mathbf{x}}$ can be expensive to compute for large neural networks, gradient updates of our method only require the Hessian-vector product (i.e., $\mathbf{H}_{\mathbf{x}}\Delta$) which can be computed efficiently (Pearlmutter, 1994). This yields the following Context-Aware Second-Order (CASO) interpretation function.

Definition 3 (The CASO Interpretation) *For a given sample* $(\mathbf{x}, y)$*, we define the Context-Aware Second-Order (CASO) importance function* $\tilde{I}^{\lambda_1,\lambda_2}_{\theta^*}(\mathbf{x}, y)$ *as follows:*

$$\tilde{I}^{\lambda_1,\lambda_2}_{\theta^*}(\mathbf{x}, y) := \max_{\Delta} \; \nabla_{\mathbf{x}} \ell\left(f_{\theta^*}(\mathbf{x}), y\right)^t \Delta + \frac{1}{2} \Delta^t \mathbf{H}_{\mathbf{x}} \Delta - \lambda_1 \|\Delta\|_1 - \lambda_2 \|\Delta\|_2^2 \tag{5}$$

We refer to the objective of this optimization as $\tilde{\ell}(\Delta)$*.* λ_1 *and* λ_2 *are defined as in* (3)*.*

3. The Impact of the Hessian

The Hessian is by definition useful when the loss function at the test sample has high curvature. However, given the linear nature of popular network architectures with piecewise linear activations, e.g., ReLU (Glorot et al., 2011) or Maxout (Goodfellow et al., 2013), *do these regions of high curvature even exist?* We answer this question for neural networks with piecewise linear activations by first providing an exact calculation of the input Hessian. Then, we use this derivation to understand the impact of including the Hessian term in interpretation. More specifically, we prove that when the probability of the predicted class is ≈ 1 and the number of classes is large, the second-order interpretation is similar to the first-order one. We verify this theoretical result experimentally over images in the ImageNet 2012 dataset (Russakovsky et al., 2015). We also observe that when the confidence in the predicted class is low, the second-order interpretation can be significantly different from the first-order interpretation. Since second-order interpretations take into account the curvature of the model, we conjecture that they are more faithful to the underlying model in these cases.

3.1. Closed-form Hessian Formula for ReLU Networks

We present an abridged version of the exact Hessian calculation here, the details are provided in Appendix A.1. Neural network models which use piecewise linear activation functions have class scores (logits) which are linear functions of the input. That is, since they are piecewise linear over the entire domain, they are linear at a particular input.[3] Thus, we can write:

$$f_\theta(\mathbf{x}) = \mathbf{W}^T \mathbf{x} + \mathbf{b},$$

where $\mathbf{x}$ is the input of dimension d, $f_\theta(\mathbf{x})$ are the logits, $\mathbf{W}$ are the weights, and $\mathbf{b}$ are the biases of the linear function. Note that $\mathbf{W}$ combines weights of different layers from the input to the output of the network. Each row $\mathbf{W}_i$ of $\mathbf{W}$ is the gradient of logit $f_\theta(\mathbf{x})_i$ with respect to the flattened

[3] Note that we ignore points where the function is non-differentiable as they form a measure zero set.

input $\mathbf{x}$ and can be handled in auto-grad software such as PyTorch (Paszke et al., 2017). We define:

$$\mathbf{p} = \text{softmax}(f_\theta(\mathbf{x}))$$
$$\ell(f_\theta(\mathbf{x}), y) = -\sum_{i=1}^{c} y_i \log(\mathbf{p}_i),$$

where c denotes the number of classes, $\mathbf{p}$ denotes the class probabilities, and $\ell(\mathbf{p}, \mathbf{y})$ is the cross-entropy loss function.

In this case, we have the following result:

Proposition 1 $\mathbf{H}_\mathbf{x}$ *is given by:*

$$\mathbf{H}_\mathbf{x} = \nabla^2_\mathbf{x} \ell(\mathbf{p}, \mathbf{y}) = \mathbf{W}(diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^T)\mathbf{W}^T \quad (6)$$

where $\text{diag}(\mathbf{p})$ is a diagonal matrix whose diagonal elements are equal to $\mathbf{p}$.

The first observation from Proposition 1 is as follows:

Theorem 2 $\mathbf{H}_\mathbf{x}$ *is a positive semidefinite matrix.*

These two results allow an extremely efficient computation of the Hessian's eigenvectors and eigenvalues using the Cholesky decomposition of $\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^T$ (Appendix C). Note the use of decomposition is critical as storing the Hessian requires intractable amounts of memory for high dimensional inputs. The entire calculation of the Hessian's decomposition for ImageNet using a ResNet-50 (He et al., 2016) runs in approximately 4.2 seconds on an NVIDIA GTX 1080 Ti.

To the best of our knowledge, this is the first work which derives the exact Hessian decomposition for piecewise linear networks. Yao et al. (2018) also proved the Hessian for piecewise linear networks is at most rank c but did not derive the exact input Hessian.

One advantage of having a closed-form formula for the Hessian matrix (6) is that we can use it to properly set the regularization parameter λ_2 in CASO's formulation. To do this, we rely on the following result:

Theorem 3 *If L is the largest eigenvalue of $\mathbf{H}_\mathbf{x}$, for any value of $\lambda_2 > L/2$, the second-order interpretation objective function (5) is strongly concave.*

We use Theorem 3 to set the regularization parameter λ_2 for CASO. We need to set λ_2 to make the optimization convex, but not set λ_2 so large that it overpowers $\mathbf{H}_\mathbf{x}$. In particular, we set $\lambda_2 = L/2 + c_1$, where we choose $c_1 = 10$ for CASO and CAFO. We observe that if c_1 is small, the optimization can become non-convex due to numerical error in the calculation of L. However above a threshold, the value of c does not have a significant impact on the saliency map.

3.2. Theoretical Results on the Hessian Impact

We now leverage the exact Hessian calculation to prove that when the probability of predicted class is ≈ 1 and the number of classes is large, the Hessian of a piece-wise linear neural network is approximately of rank one and its eigenvector is approximately parallel to the gradient. Since a constant scaling does not affect the visualization, this causes the two interpretations to be similar to one another.

Theorem 4 *If the probability of the predicted class=1-(c-1)ϵ , where $\epsilon \approx 0$, then as $c \to \infty$ such that $c\epsilon \approx 0$, Hessian is of rank one and its eigenvector is parallel to the gradient.*

Let Δ^*_{CASO} be the optimal solution to the CASO objective 5 and Δ^*_{CAFO} be the optimal solution for the CAFO objective 3. We assume λ_1=0 for both the objectives.

Theorem 5 *If the probability of the predicted class=1-(c-1)ϵ , where $\epsilon \approx 0$, then as $c \to \infty$ such that $c\epsilon \approx 0$, the CASO solution (5) with $\lambda_1 = 0$ is almost parallel to the CAFO solution (3) with $\lambda_1 = 0$.*

We emphasize that our theoretical results are valid in the "asymptotic regime". To analyze the approximation in the finite length regime, we simulate the relative error between the true Hessian and the rank-one approximation of the Hessian as the number of classes increases and probability of predicted class tends to 1. We find the Hessian quickly converges to rank-one empirically (Appendix F.1).

3.3. Empirical Results on the Hessian Impact

We now present empirical results on the impact of the Hessian in interpreting deep learning models. In our experiments here, we isolate the impact of the Hessian term by setting $\lambda_1 = 0$ in both CASO and CAFO.

A consequence of Theorem 3 is that the gradient descent method with Nesterov momentum converges to the global optimizer of the second-order interpretation objective with a convergence rate of $\mathcal{O}(1/t^2)$, see Appendix B for details.

To optimize Δ, the gradient is given by:

$$\nabla_\Delta \tilde{\ell}(\Delta) = \nabla_\mathbf{x} \ell\left(f_{\theta^*}(\mathbf{x}), y\right) + \mathbf{H}_\mathbf{x}\Delta - 2\lambda_2\Delta. \quad (7)$$

The gradient term $\nabla_\mathbf{x} \ell\left(f_{\theta^*}(\mathbf{x}), y\right)$ and the regularization term $-2\lambda_2\Delta$ are straightforward to implement using standard backpropagation.

To compute the Hessian-vector product term $\mathbf{H}_\mathbf{x}\Delta$, we rely on the result of Pearlmutter 1994 (Pearlmutter, 1994): a Hessian-vector product can be computed in the same time as the gradient $\nabla_\mathbf{x} \ell\left(f_{\theta^*}(\mathbf{x}), y\right)$. This is handled easily in modern auto-grad software. Moreover, for ReLU networks, our closed-form formula for the Hessian term (Theorem

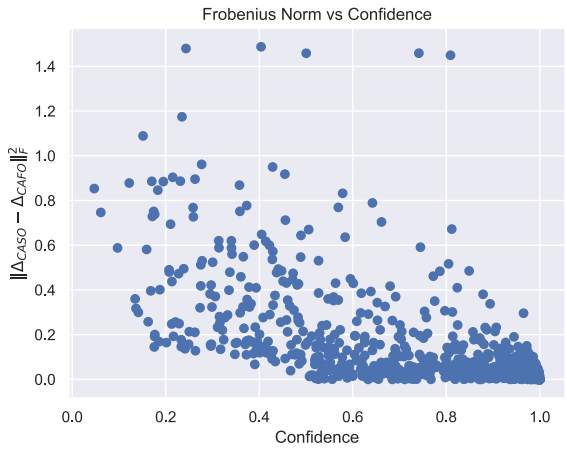


Figure 1. The Frobenius norm difference between CASO and CAFO after normalizing both vectors to have the same L_2 norm. Consistent with the result of Theorem 4, when the classification confidence is low, the CASO result differs significantly from CAFO. When the confidence is high, CASO and CAFO are approximately the same. To isolate the impact of the Hessian term, we assume $\lambda_1 = 0$ in both CASO and CAFO.

1) can be used in the computation of the Hessian-vector product as well. In our experiments here we use the closed-form formula for $\lambda_1 = 0$. When $\lambda_1 > 0$, we use proximal gradient descent (Section 4).

We compare second-order (CASO with $\lambda_1 = 0$) and the first-order interpretations (CAFO with $\lambda_1 = 0$) empirically. Note that when $\lambda_1 = 0$, $\Delta_{CAFO} = \frac{1}{\lambda_2}\mathbf{g_x}$ where $\mathbf{g_x}$ is the gradient and Δ_{CAFO} is the interpretation obtained using the CAFO objective.

We compute second-order and first-order interpretations for 1000 random samples on the ImageNet ILSVRC-2012 (Russakovsky et al., 2015) validation set using a Resnet-50 (He et al., 2016) model. Our loss function $\ell(\cdot, \cdot)$ is the cross-entropy loss. After calculating Δ for all methods, the values must be normalized for visualization in a saliency map. We apply a normalization technique from existing work which we describe in Appendix D.

We plot the Frobenius norm of the difference between CASO and CAFO in Figure 1. Before taking the difference, we normalize the Δ solutions produced by CASO and CAFO to have the same L_2 norm because a constant scaling of elements of Δ does not change the visualization.

The empirical results are consistent with our theoretical results: second-order and first-order interpretations are similar when the classification confidence is high. However, when the confidence is small, including the Hessian term can be useful in deep learning interpretation.

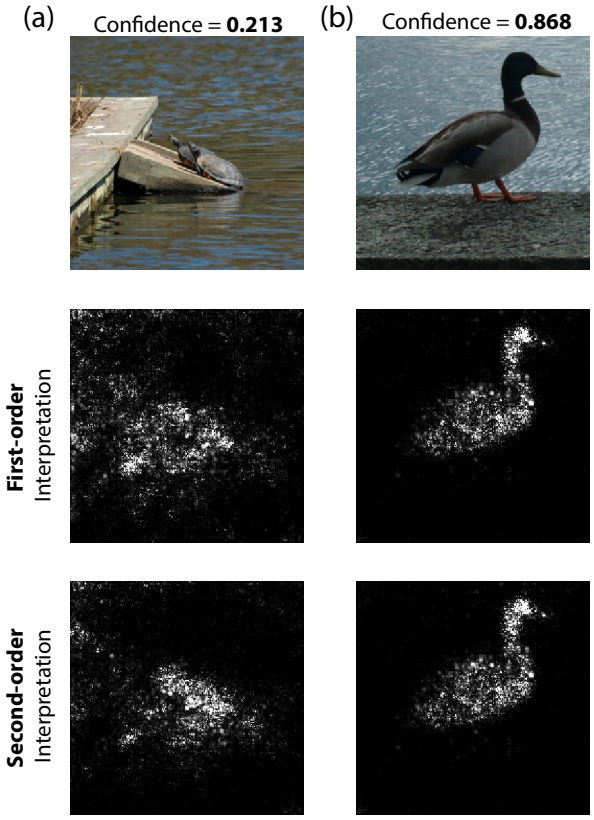


Figure 2. Panel (a) shows an example where the classification confidence is low. In this case, the CASO and CAFO interpretations differ significantly. Panel (b) demonstrates an example where the classification confidence is high. In this case, CASO and CAFO lead to similar interpretations as suggested by our theory.

To observe the difference between CAFO and CASO interpretations qualitatively, we compare them for an image when the confidence is high and for one where it is low in Figure 2. When the classification confidence is high, CAFO $\approx$ CASO and when this is low, CASO $\neq$ CAFO. Additional examples have been given in Appendix F.

We do additional experiments to evaluate the impact of the Hessian on a neural network that is not piecewise linear. We interpret a SE-Resnet-50 (Hu et al., 2018) neural network (which uses sigmoid non-linearities) on the same 1000 images. We observe a similar trend as in the case of ReLU networks (Appendix F.3).

4. The Impact of Group-features

This section studies the impact of the group-features in deep learning interpretation. The group-feature has been included as the sparsity constraint in optimization (2).

To obtain an unconstrained concave optimization for the CASO interpretation, we relaxed the sparsity (cardinality) constraint $\|\Delta\|_0 \leq k$ (often called an L_0 norm constraint) to a convex L_1 norm constraint. Such a $L_0 - L_1$ relaxation is a core component for popular learning methods such as compressive sensing (Candes & Tao, 2005; Donoho, 2006) or LASSO regression (Tibshirani, 1996). Using results from this literature, we show this relaxation is tight under certain conditions on the Hessian matrix $\mathbf{H_x}$ (see Appendix E). In other words, the optimal Δ of optimization (5) is sparse with the proper choice of regularization parameters.

Note that the regularization term $-\lambda_1\|\Delta\|_1$ is a concave function for $\lambda_1 > 0$. Similarly due to Theorem 3, the CASO interpretation objective (5) is strongly concave.

One method for optimizing this objective is using gradient descent as done in the second-order interpretation but using an L_1 regularization penalty. However, we found that this procedure leads to poor convergence properties in practice, partially due to the non-smoothness of the L_1 term.

To resolve this issue, we instead use *proximal* gradient descent to compute a solution for CAFO and CASO when $\lambda_1 > 0$. Using the Nesterov momentum method and backtracking with proximal gradient descent gives a convergence rate of $\mathcal{O}(1/t^2)$ where t is the number of gradient updates (Appendix B).

Below we explain how we use proximal gradient descent to optimize our objective. First, we write the objective function as the sum of a smooth and non-smooth function:

$$\tilde{\ell}(\Delta) = \underbrace{\nabla_{\mathbf{x}}\ell\left(f_{\theta^*}(\mathbf{x}), y\right)^t \Delta + \frac{1}{2}\Delta^t \mathbf{H_x}\Delta - \lambda_2\|\Delta\|_2^2}_{\text{Smooth Part}} \underbrace{-\lambda_1\|\Delta\|_1}_{\text{Non-Smooth Part}}$$

Let $g(\Delta)$ be the smooth, $h(\Delta)$ be the non-smooth part:

$$\tilde{\ell}(\Delta) = g(\Delta) + h(\Delta)$$

$$g(\Delta) = \nabla_{\mathbf{x}}\ell\left(f_{\theta^*}(\mathbf{x}), y\right)^t \Delta + \frac{1}{2}\Delta^t \mathbf{H_x}\Delta - \lambda_2\|\Delta\|_2^2$$

$$h(\Delta) = -\lambda_1\|\Delta\|_1$$

The gradient of the smooth objective is given by:

$$\nabla_\Delta g(\Delta) = \nabla_{\mathbf{x}}\ell\left(f_{\theta^*}(\mathbf{x}), y\right) + \mathbf{H_x}\Delta - 2\lambda_2\Delta$$

The proximal operator is given by:

$$\text{prox}_\alpha(x) = \arg\min_z \frac{1}{\alpha}\|x - z\|_2^2 + \lambda_1\|z\|_1$$

$$= \begin{cases} x + \lambda_1\alpha & x \leq -\lambda_1\alpha \\ 0 & -\lambda_1\alpha < x \leq \lambda_1\alpha \\ x - \lambda_1\alpha & \lambda_1\alpha < x \end{cases}$$

This formula can be understood intuitively as follows. If the magnitude of some elements of Δ is below a certain threshold ($\lambda_1\alpha$), proximal mapping sets those values to zero. This leads to values that are exactly zero in the saliency map.

To optimize Δ, we use FISTA (Beck & Teboulle, 2009) with backtracking and the Nesterov momentum optimizer with a learning rate of 0.1 for 10 iterations and decay factor of 0.5. Δ is initialized to zero. FISTA takes a step with learning rate α to reduce the smooth objective loss $g(\Delta)$ and then applies a proximal mapping to the resulting Δ. Backtracking reduces the learning rate when the update results in a higher loss.

4.1. Empirical Impact of Group-Features

We now investigate the empirical impact of group-features. In our experiments, we focus on image classification because visual interpretations are intuitive and allow for comparison with prior work. We use a Resnet-50 (He et al., 2016) model on the ImageNet ILSVRC-2012 dataset.

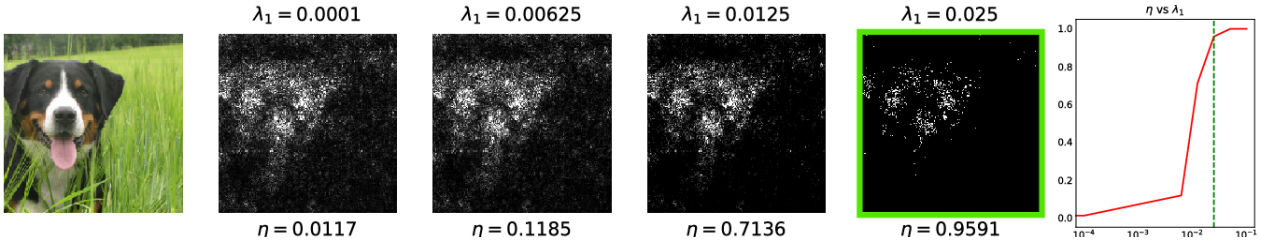
To gain an intuition for the effect of λ_1, we show a sweep over values in Figure 3. When λ_1 is too high, the saliency map becomes all zero. Different approaches to set the regularization parameter λ_1 have been explored in different problems. For example, in LASSO, one common approach is to use Least Angle Regression (Efron et al., 2004).

We propose an unsupervised method based on the sparsity ratio of the interpretation solution to set λ_1. We define η, the sparsity ratio, as the number of zero pixels divided by the total number of pixels. We start with $\lambda_1 = 10^{-5}$ and increase λ_1 by a factor of 10 until Δ reaches all zeros. For interpretations with sparsity in a certain range (e.g. $1 > \eta \geq 0.75$ in our examples), we choose the interpretation with the highest loss. If we do not find any interpretation that satisfies the sparsity condition, we reduce the first λ_1 that resulted in Δ becoming zero by a factor of 2 and repeat further iterations. In practice, we batch different values of λ_1 to find a reasonable parameter setting efficiently.

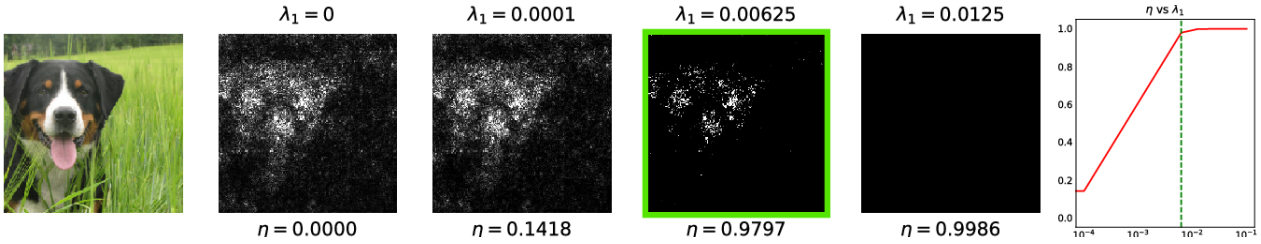
This method selects the interpretation marked with a green box in Figures 3a and 3b. In Figure 3c, we show the gradient interpretation with different values of clipping thresholds to induce the specified sparsity value. We observe that the interpretations obtained using group-features (Figures 3a and 3b) are less noisy compared to Figure 3c.

5. Qualitative Comparision of Deep Learning Interpretation Methods

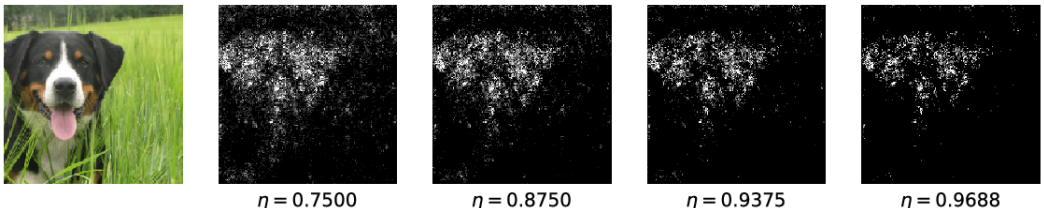
This section briefly reviews prior saliency map approaches and compares their performance to CAFO and CASO qualitatively. The proposed Hessian and group-feature terms can be included in existing approaches as well.



(a) Interpretation solutions for CAFO with different values of the regularization parameter λ_1.



(b) Interpretation solutions for CASO with different values of the regularization parameter λ_1.



(c) Interpretation solutions using the gradient with different clipping thresholds to induce the given sparsity.

Figure 3. Larger λ_1 values lead to higher sparsity ratios (η). Our unsupervised method selects the interpretations marked with a green box. Interpretations selected in panel (a) and (b) are less noisy compared to (c).

Vanilla Gradient: Simonyan et al. (2014) propose to compute the gradient of the class score with respect to the input.

SmoothGrad: Smilkov et al. (2017) argue that the input gradient may fluctuate sharply in the region local to the test sample. To address this, they average the gradient-based importance values generated from many noisy inputs.

Integrated Gradients: Sundararajan et al. (2017) define a baseline, which represents an input absent of information (e.g., a completely zero image). Feature importance is determined by accumulating gradient information along the path from the baseline to the original input: $(\mathbf{x}-\mathbf{x}') \times \int_{\alpha=0}^{1} \nabla_{\mathbf{x}} \ell\left(f_{\theta^*}(\mathbf{x}' + \alpha(\mathbf{x}-\mathbf{x}')), y\right)\, d\alpha$. The integral is approximated by a finite sum.

We use the normalization method from SmoothGrad (Smilkov et al., 2017) for visualizing the saliency map. Details of this method are given in Appendix D.

We can also extend the idea of SmoothGrad to define smooth versions of CASO and CAFO. This yields the following interpretation objective.

Definition 4 (The Smooth CASO Interpretation) *For a given sample $(\mathbf{x}, y)$, we define the smooth context-aware second-order (the Smooth CASO) importance function $\tilde{I}_{\theta^*}^{\lambda_1,\lambda_2}(\mathbf{x}, y)$ as follows:*

$$\tilde{I}_{\theta^*}^{\lambda_1,\lambda_2}(\mathbf{x}, y) := \max_{\Delta} \; \frac{1}{n}\sum_{1}^{n}\Big(\nabla_{\mathbf{z}}\ell\left(f_{\theta^*}(\mathbf{z}), y\right)^t \Delta + \frac{1}{2}\Delta^t \mathbf{H}_{\mathbf{z}} \Delta\Big) - \lambda_1\|\Delta\|_1 - \lambda_2\|\Delta\|_2^2 \tag{8}$$

where $\mathbf{z} = \mathbf{x} + N(0, \sigma^2 I)$ and λ_1 and λ_2 are defined similarly as before.

In the smoothed versions, we average over $n = 50$ samples with $\sigma = 0.15$. Smooth CAFO is defined similarly without the Hessian term.

Since quantitatively evaluating a saliency map is an open problem, we focus on two qualitative aspects. First, we inspect visual coherence, i.e., only the object of interest should be highlighted and not the background. Second, we test for discriminativity, i.e., in an image with two objects the predicted object should be highlighted.

Figure 4 shows comparisons between CAFO, CASO, and

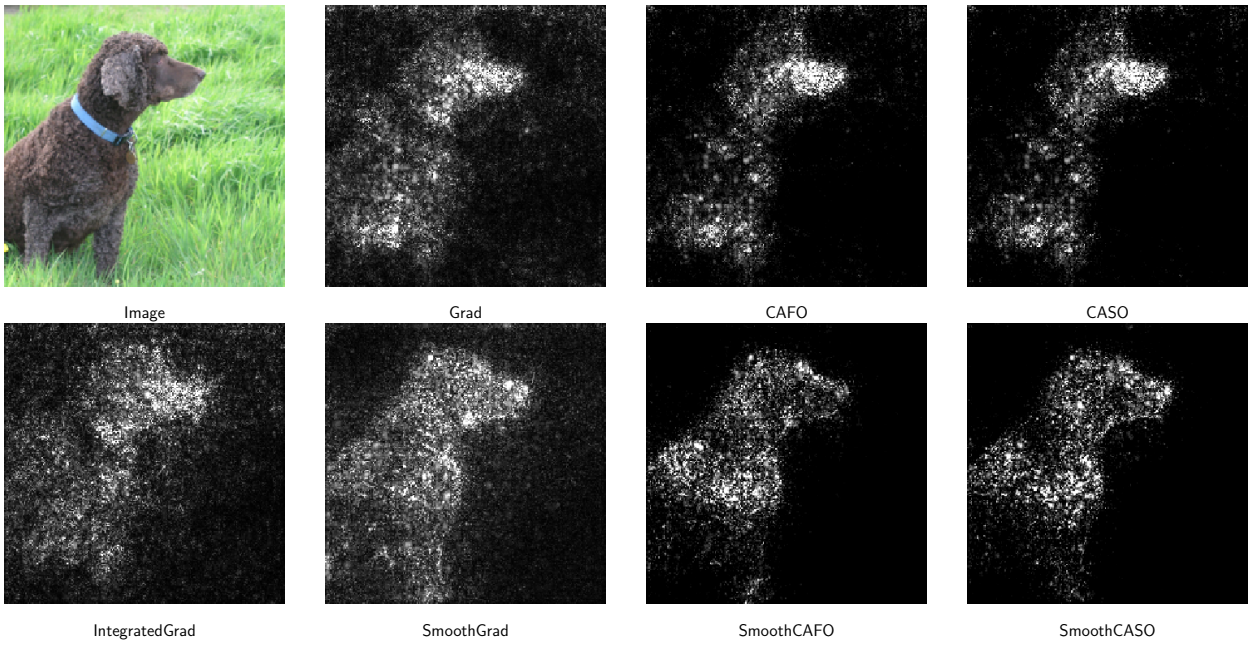


Figure 4. A qualitative comparison of existing interpretation methods. More examples are shown in Appendix H. Grad stands for Vanilla Gradient and IntegratedGrad stands for Integrated Gradient. For our methods (CAFO, CASO, SmoothCAFO, SmoothCASO) the saliency map is more visually coherent with the object of interest compared to existing methods.

other existing interpretation methods. Including group-features in the interpretation leads to a sparse saliency map, eliminating the spurious noise and creating a visually coherent saliency map. More examples have been presented in Appendix H.

6. Conclusion and Future Work

We have studied two aspects of the deep learning interpretation problem. First, we characterized a closed-form formula for the input Hessian matrix of a deep ReLU network. Using this, we showed that, if the confidence in the predicted class is high and the number of classes is large, first-order and second-order methods produce similar results. In the process, we also proved that the Hessian matrix is of rank one and its eigenvector is parallel to the gradient. These results can be insightful in other related problems such as adversarial examples. Second, we incorporated feature interdependencies in the interpretation using a sparsity regularization term. Adding this term significantly improves qualitative interpretation results.

There remain many open problems in interpreting deep learning models. For instance, since saliency maps are high-dimensional, they can be sensitive to noise and adversarial perturbations (Ghorbani et al., 2019). Moreover, without proper quantitative evaluation metrics for model interpretations, the evaluation of interpretations is often qualitative and can be subjective. Finally, the theoretical impact of the Hessian term for low confidence predictions and the case when the number of classes is small remains unknown. Resolving these issues are among interesting directions for future work.

Acknowledgments

Shi Feng and Eric Wallace were supported by NSF Grant IIS-1822494. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsor.

References

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2009.

Candes, E. J. and Tao, T. Decoding by linear programming. *IEEE transactions on information theory*, 2005.

Donoho, D. L. Compressed sensing. In *IEEE Transactions on Information Theory*, 2006.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least Angle Regression. *The Annals of Statistics*, June 2004.

Feng, S., Wallace, E., II, A. G., Iyyer, M., Rodriguez, P., and Boyd-Graber, J. Pathologies of neural models make interpretations difficult. In *Proceedings of Empirical Methods in Natural Language Processing*, 2018.

Ghorbani, A., Abid, A., and Zou, J. Y. Interpretation of neural networks is fragile. *AAAI*, 2019.

Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of Artificial Intelligence and Statistics*, 2011.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. Maxout networks. In *Proceedings of the International Conference of Machine Learning*, 2013.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, 2016.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Li, J., Monroe, W., and Jurafsky, D. Understanding neural networks through representation erasure. *arXiv preprint arXiv: 1612.08220*, 2016.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations*, 2018.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. DeepFool: a simple and accurate method to fool deep neural networks. *Computer Vision and Pattern Recognition*, 2016.

Nie, W., Zhang, Y., and Patel, A. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. *International Conference on Machine Learning*, 2018.

Parikh, N. and Boyd, S. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, January 2014. ISSN 2167-3888. doi: 10.1561/2400000003. URL http://dx.doi.org/10.1561/2400000003.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques*, 2017.

Pearlmutter, B. A. Fast exact multiplication by the hessian. In *Neural Computation*, 1994.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations*, 2014.

Smilkov, D., Thorat, N., Kim, B., Viégas, F. B., and Wattenberg, M. SmoothGrad: removing noise by adding noise. *arXiv preprint arXiv: 1706.03825*, 2017.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the International Conference of Machine Learning*, 2017.

Tibshirani, R. Regression shrinkage and selection via the lasso. In *Journal of the Royal Statistical Society*, 1996.

Yao, Z., Gholami, A., Lei, Q., Keutzer, K., and Mahoney, M. W. Hessian-based analysis of large batch training and robustness to adversaries. *Neural Information Processing Systems*, 2018.