L_1-Norm Gradient Penalty for Noise Reduction of Attribution Maps

Keisuke Kiritoshi
NTT Communications
Tokyo, Japan
k.kiritoshi@ntt.com

Ryosuke Tanno
NTT Communications
Tokyo, Japan
r.tanno@ntt.com

Tomonori Izumitani
NTT Communications
Tokyo, Japan
tomonori.izumitani@ntt.com

Abstract

Determining the attribution of the input elements to the output values is very important for interpretability when we use deep neural network (DNN) models in real-world tasks. Gradient-based methods are widely used because they can represent the relationship between each input and output pair in the shape of a partial derivative. Attribution values determined from DNN models that use batch normalization include high levels of noise. This is problematic because it significantly reduces the interpretability of the model. To obtain sparse and interpretable attribution maps, we developed a new regularization method that includes a penalty term, based on the L_1-norm of gradient values calculated through back-propagation procedures, in the loss function. We evaluated the effectiveness of the method using CIFAR-10 image datasets.

1. Introduction

In the machine learning field, explaining why neural network models make decisions and predictions is very important. Understanding why a model gives a specific answer is crucial for reliability and safety, especially when deploying neural network technology in different industries, medical facilities, and factories.

There have been several studies that have demonstrated the relationship between the input and output of trained neural network models (*Attributions*, and we call its [value — output?] an *attribution map*). Backpropagation-based methods use partial differentials of the output with respect to the model input in order to extract the attributions. They visualize which pixels models focus on for the attribution maps.

Backpropagation-based methods are separated into two types: vanilla gradient-based, and extended gradient-based. The former allows us to analyze the relationship between the input and output based on the nature of gradients themselves [4], unlike the latter.

In this study, we focused on vanilla gradient based methods, and we showed batch normalization tends to add a high level of noise to attribution maps. We developed a new regularization method named L_1*-Norm Gradient Penalty* to remove the noise from attribution maps. This method regularized attribution maps during the training phase to make attribution maps sparser. To evaluate the effectiveness of our method, we carried out image classification experiments and showed that the noise in the attribution maps had decreased.

Our Contributions: (1) We showed that batch normalization affects noise levels in attribution maps extracted by vanilla gradient methods. (2) We used a L_1*-Norm Gradient penalty* to reduce the noise caused by batch normalization without affecting the accuracy, and we evaluated the effectiveness of our method with additional experiments.

2. Related Works

The attribution map between the input and output of neural networks has been studied with regards to visualization and interpretability. A typical approach is to run a back-propagation algorithm with models. These methods fall into two categories defined by how the attributions are calculated.

Vanilla Gradient-Based Method: This method extracts attribution maps from the original gradients of the output with respect to input?. Simonyan et al. extracted the original gradient values of the output with respect to the input to produce attributions of neural networks [6]. Sundararajan et al. [9] derived their integrated gradient method from the saliency map method. They focused on satiating the output of the model and showed that this lowered the accuracy of the saliency map. Smoothgrad [8] removes noise from attribution maps by calculating the average gradients of some noise-added samples. Time-smoothgrad [4] extends the Smoothgrad concept to time-series regression tasks and reduces noise in attribution maps by comparing them with vanilla gradients.

Extended Gradient-Based Method: In this method, the attribution maps are calculated by using different equations based on gradients. Binder et al. [2] defined a "relevance

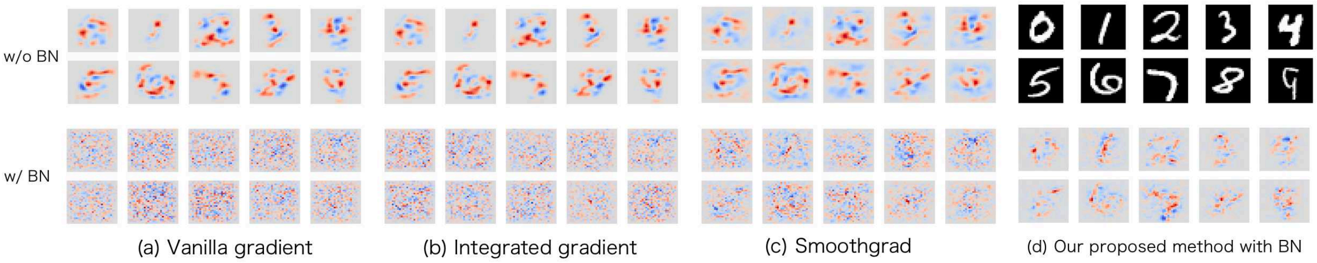

Figure 1. Comparison of Attribution Maps: Column (a): vanilla gradient, column (b): Integrated Gradient, column (c): Smoothgrad, and column (d): our method with batch normalization. Each attribution map from the model with batch normalizations (the left side) was incapable of representing the shapes of the numbers in the original images on the right. On the other hand, the attribution maps in the center that didn't use batch normalization are capable of showing rough outlines of the numbers. As a result, we deduced that batch normalization causes higher noise levels in attribution maps that have been extracted by using gradients.

score" that reflects the importance of each node in a neural network; the sum for each layer is equal to the output values. The relevance score is calculated by using back-propagation of the output values. DeepLIFT [5] is an extension of this method that consider the effect of the plus or minus sign of the relevance score for each node.

Building on these studies, our research focuses on the vanilla gradient method and reveals that batch normalization causes high noise levels in attribution maps. We developed a new regularization for the training loss function. Whereas the related studies focus on how to calculate attributions without changing trained models, our method produces models that are trained to generate sparse attribution maps instead of calculating new attribution maps.

3. Proposed Method

3.1. Noise of Attribution Map by Batch Normalization

To investigate how batch normalization affects attribution maps, we carried out a simple experiment with a MNIST dataset. Figure 1 shows a comparison between the simple gradients of the output and input with and without batch normalization. The network consisted of two convolutional layers and one full connected layer and batch normalization layers were inserted in the convolution layers. We compared the saliency map [6], integrated gradient [9], and smoothgrad [8]. Each method resulted in attribution maps with high noise levels in the case of the models that used batch normalization layers.

We examined the relationship between model input and output, including those with batch normalization layers as per the following procedure. Here, we define *Model'* as a trained model in which the last softmax layer has been removed.

(1) Extract zero pixel indices in all test data samples.

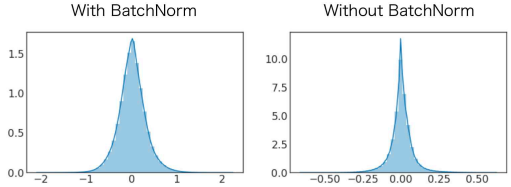

Figure 2. Distribution of Difference between Input and Output

(2) Select one index from the indices (1) and create a sample converted value of the index into one.

(3) Feed sample (2) to *Model'* and obtain output values.

(4) Repeat (2) and (3) for each index (1) and calculate the difference between the output values of the original test samples and the output value list provided by the steps above.

Figure 2 shows the distribution of the difference. The distribution based on the model with batch normalization varies more than the distribution without batch normalization. This result infers the output images are more affected by unnecessary input pixels in models with batch normalizations, and this phenomenon can cause noise in attribution maps.

3.2. L_1 Norm Gradient Penalty

To reduce noise in attribution maps for image classification tasks, we introduce a new regularized L_1-norm gradient penalty to the loss functions L' of neural networks as follows:

$$L' = L(\boldsymbol{x}, \boldsymbol{y}) + \alpha \left| \frac{\partial S_c(\boldsymbol{x})}{\partial \boldsymbol{x}} \right| \quad (1)$$

Here, $L((\boldsymbol{x}, \boldsymbol{y})$ is the original loss function and $S_c(\boldsymbol{x})$ is the output of the layer immediately before the final softmax layer of the true class c. α is the hyper-parameter to decide the size of the regularization. The second term on the right side means L_1 is the norm of the simple gradients of

Method	VGG16	VGG19
Baseline	0.713	0.730
L_1 Gradient Penalty	**0.730**	**0.743**

Table 1. Accuracy for CIFAR-10 dataset.

output with respect to the neural network input proposed by Simonyan *et al*.

We hoped to get regular and sparse attribution maps directly in the training phase. Generally, not all input features contribute to classifications or predictions. For example, in image classification tasks such as CIFAR-10 and Imagenet, subjects which models should recognize might be seen in only part of the image. In many cases, models do not need many background pixels, so the background of the attribution maps should be sparse.

The idea of regularizing gradients of output with respect to input is similar to the regularization of WGAN-GP [3]. WGAN-GP uses the L_2 norm for weight clipping to maintain the gradient norm, while our method uses the L_1 norm to calculate the gradients of sparse attribution maps.

4. Experiments

4.1. Qualitative Evaluation

To determine whether noise could be eliminated by using our method, we visualized attribution maps from models trained with the L_1-norm gradient penalty and the baseline (without a penalty). We trained the VGG16 and VGG19 architectures [7] on the CIFAR-10 dataset and each architecture included batch normalization as the next layer of each convolution layer. Note that the models are not pre-trained by any datasets.

Figure 3 shows examples of attribution maps from VGG16 model trained with our method and the baseline. We extracted attribution maps by using (a) the vanilla gradient by Simonyan *et al*. [6], (b) the integrated gradient [9], (c) Smoothgrad [8], and (d) our method. Table 1 shows the accuracy of each method.

4.2. Modified Sensitivity-n Evaluation

There is no common metric to evaluate attribution scores and few studies have been made on this problem so far. Ancona *et al*. proposed a metric Sensitivity-n based on relationship between attribution scores of feature subsets and output variations[1]. We use this metric, with minor modification, for quantitative evaluation.

The Sensitivity-n metric is based on the idea that if contribution of a feature subset is large, the sum of their attribution scores Rn(x) tends to be proportional to the output score variation $S_n(x)$. It utilize the Pearson correlation coefficient between $R_n(x)$ and $S_n(x)$ as the metric.

For an input x, $R_n(x)$ and $S_n(x)$ are calculated as follows:

$$R_n(x) = \sum_{i=1}^{n} R_i^c(x) \tag{2}$$

$$S_n(x) = S_c(x) - S_c(x_{[x_S=0]}) \tag{3}$$

$$x_S = [x_1, x_2, \ldots x_n] \subseteq x \tag{4}$$

Here, $R_i^c(x)$ and $S_c(x)$ are attribution score of feature x_i regarding output class c and output value regarding class c, respectively. In the experimental setting Ancona *et al*. [1], Sensitivity-n is prepared as following steps: (1) Randomly sample 100 subsets x_S of features from input x of size n and calculate $R_n(x), S_n(x)$ for each x_S. (2) Calculate Pearson correlation coefficient by the sequences $R_n(x)$ and $S_n(x)$. (3) Carry out step (1) and (2) for 1000 test samples (4) Compare sample the averaged correlation while changing n from 1 to 1000.

Note that in our experiment, we modified correlation function and use absolute Pearson correlation coefficient instead of normal Pearson correlation coefficient because training data is standardized as preprocessing. When training data includes plus and minus values, the correlation can be minus even if attribution values contribute the output variations.

Figure 4 shows the variation of the averaged absolute correlation for each n of each training method. Compared with the baseline method, our method resulted in a higher correlations for each training procedure on both VGG16 and VGG19.

5. Discussion

Interplitability and Accuracy

As shown in Figure 3, the models trained with the L_1-norm Gradient Penalty sparse background pixels showed higher attribution values for important pixels on the subjects compared with the other attribution methods. Table 1 shows that our method did not decrease accuracy. We can conclude our regularization creates more interpretable models without sacrificing accuracy.

Effectiveness of Attribution

As shown in Figure 4, the model trained with L_1-norm Gradient Penalty keeps higher correlation of the attribution and the output variation than model without any regularizations. This result means the attribution map of the model with the proposed method can represent the contribution the output directly.

6. Conclusion

In this study, we addressed the high noise levels in attribution maps caused by batch normalization (especially in vanilla gradient methods). We used the MNIST dataset to

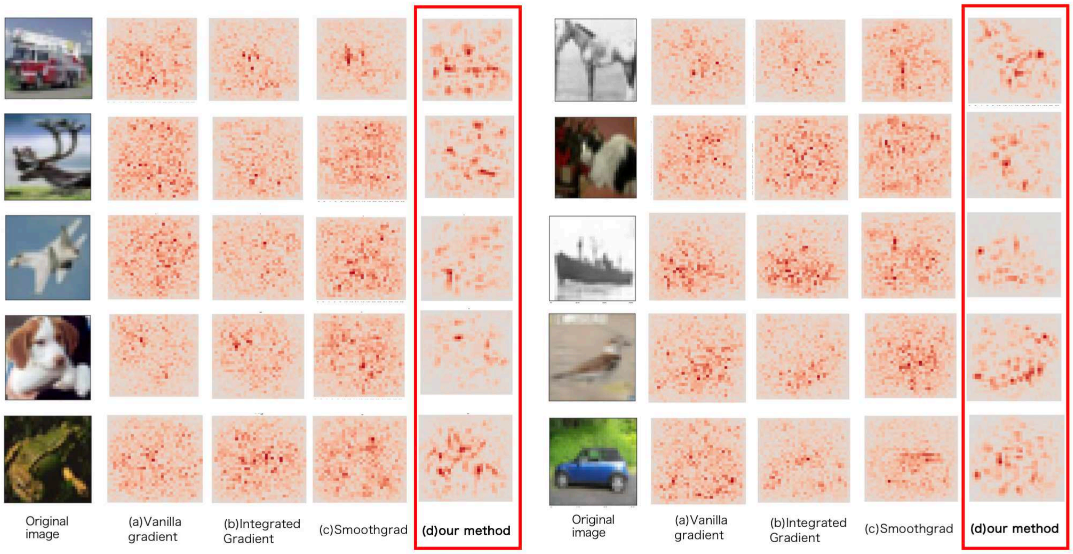

Figure 3. Comparison of Attribution Maps: (a) vanilla gradient, (b) Integrated Gradient, (c) Smoothgrad, and (d) our method. The attribution maps are the average absolute values of the input channel direction (Red: +). By focusing on unimportant background pixels around the subjects, our method makes them sparser than the other methods can.

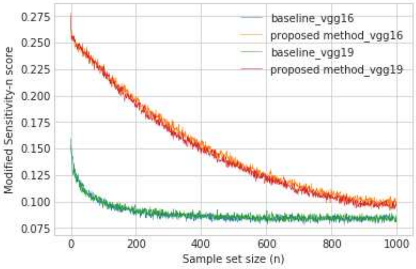

Figure 4. Modified Sensitivity-n Evaluation: the vertical and horizontal axes respectively show the averaged absolute correlation and n.

conduct our experiments. To reduce and sparsify noise, we used a L_1-norm gradient penalty, which regularized the attribution maps directly during the training phase. In our study, attribution maps calculated using our method had lower levels of background noise compared with those calculated using other methods.

In future, we hope to find out why models that use batch normalizations cause high noise levels in attribution maps and why the L_1-norm gradient penalty can reduce the noise. In this study, we only carried out experiments on simple convolutional neural network models; we need to confirm the effectiveness of the method for other models, such as multilayer perceptron and recurrent neural networks.

References

[1] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. *International Conference on Learning Representations (ICLR 2018)*, 2018.

[2] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek. Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers. In *Artificial Neural Networks and Machine Learning*, pages 63–71. Springer, 2016.

[3] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved Training of Wasserstein GANs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017.

[4] K. Kiritoshi, K. Ito, and T. Izumitani. Capturing Time-Varying Influence Using an Attribution Map Method for Neural Networks. *IJCAI Workshop on AI for Internet of Things(AI4IoT)*, 2018.

[5] A. Shrikumar, P. Greenside, and A. Kundaje. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3145–3153, 2017.

[6] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *In arXiv preprint arXiv:1312.6034*, 2013.

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[8] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *ICML Workshop on Visualization for Deep Learning*, 2017.

[9] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328, 2017.