Towards explaining gradient-based visualizations in deep neural networks

Weili Nie

October 24, 2017

Abstract

Interpretability of deep neural networks is essential to open the black box of deep learning. Some approaches based on gradients have been proposed to get compelling visualizations, but there is a lack of theory to explain what we have observed in experiments. Motivated by this, we propose a theoretical explanation for these visualizations in deep nerual networks and reveal that it is the local connections in any convnet that result in human interpretable visualizations no matter the network has been trained or not. Our results suggest that gradient-based visualization mehods convey little information about the value of weights and the importance of pixels to a specific class. Finally, our experiments verify theoretical analysis.

1 Introduction

Deep neural networks are considered as a black box despite of their extensive applications [3, 5, 9]. To make them more interpretable, many visualization techniques [6, 10, 8]have been proposed by highlighting task-relevant pixels (i.e. the intensity changes of these pixels have the most significant impact on the individual decisions of classifiers). [6] visualized the spatial support of a given class in a given image, i.e. plain saliency map, by using the true gradient. [10] visualized the mapping from feature activities back to the input pixel space with a deconvolutional network (deconvnet). [8] proposed to combine these two methods: rather than masking out values corresponding to negative entries of the top gradient (deconvnet) or bottom data (plain saliency map), they masked out the values for which at least one of these values is negative and produced sharper visualizations.

This class of gradient-based visualizations has attracted a lot attentions in the deep learning community []. But the reason why gradient-based visualizations can produce visually interpretable images is not understood well. [6] proposed two possible explanations for gradient-based visualizations. First, approximating the highly nonlinear neural networks with a linear function

$$f_k(x) \approx w^T I + b$$

in the neighborhood of a given image I_0, the plain gradient of network output over input is just the weights we learned. In such sense, we are actually visualizing the *learned weights* by doing gradient-based visualizations. Later on, [2] extended it and proposed a more complicated visualization method called PatternNet based on their analsysis about linear models. Second, the magnitude of class score derivative can be used to indicate which pixels need to be changed the least to affect the specific class score the most, and such pixels correspond to object location in the image.

However, our recent experimental results are contradictory to both of the above explanations, which causes some open questions: Does this class of visualization methods really reveal the mystery of neural networks? How to justify gradient-based visualizations?

2 Empirical Observations

First, we randomly choose an image from Imagenet dataset [1] and apply guided backpropagation to two models: (1) an untrained VGG-16 net [7] with weights being randomly initialized, (2) a well-trained VGG-16 net. Noe that throughout this paper, we calculate guided backpropagation for one of the logits (i.e. the output right before softmax function) instead of the cost function without loss of generality. Figure 1 shows the visualizations for untrained and trained models.

input

(a) trained-max (b) trained-482 (c) trained-560

(d) init-max (e) init-731 (f) init-815

Figure 1: Guided backpropagation visualizations for both trained VGG-16 net (a)-(c) and untrained VGG-16 net (d)-(f) given an input "tabby" from Imagenet, where "max" refers to computing gradient for the maximum logit and the number, say "482", refers to computing gradient for the 482-th logit. These numbers in the figure are randomly chosen for generality.

As we can see from Figure 1 that both trained and untrained VGG-16 net could produce a human interpretable guided backpropagation visualizations

with a subtle difference: it cares mode about edges in the trained model while it turns to recover the original image as a whole in the untrained model. As predicted by the "learned weights" explanation based a linear function, however, the guided backpropagation in the untrained VGG-16 net should be random noise as the weights here are all randomly sampled from the Gaussian distribution. Therefore, the analysis based on linear models is not able to explain the these behaviors of neural networks and might be misleading in some cases.

Furthermore, we observe that the guided backpropagation is invariant to which logit we choose. Though having not shown here, we also observe the class invariance phenomena in other gradient-based visualizations, such as plain salency map and deconvnet. It means that this class of visualizations conveys little information about making a decision for classification. This result clearly contradicts to the previous explanation that the gradient-based visualizations are the indicator of important pixels to a specific class.

3 Theoretical Explanations

In this section, we start from a very simple convoluntional neural network (convnet) to propose an analytic explanation to the above observations.

Consider a three-layer convnet, consisting of a convolutional layer, followed by a ReLU activation function and a fully connected layer of which output is called logits. Formally, let $x \in \mathbb{R}^d$ be an input image and $W \in \mathbb{R}^{p\times N}$ be N convolutional filters where each column $w^{(i)}$ denotes a filter with size p. Then we let $Y \in \mathbb{R}^{p\times J}$ be J patches from x, and each column $y^{(j)}$ with size p is generated by some liner function $y^{(j)} = D_j x$ where $D_j = \begin{bmatrix} 0_{p\times(j-1)b} & I_{p\times p} & 0_{p\times(d-(j-1)b-p)} \end{bmatrix}$ with b being the stride size [1]. For a filter with size 3 and stride 1, then $y^{(j)}$ is the j-th to $(j+2)$-th consecutive pixels. The weights in the fully-connected layer can be denoted by $V \in \mathbb{R}^{NJ\times K}$ with K being the number of output logits. Denote by $\sigma(x) = \max(x, 0)$ the ReLU activation function, the k-th logit is represented by

$$f_k(x) = \sum_{i=1}^{N}\sum_{j=1}^{J} V_{q_{ij},k}\sigma({w^{(i)}}^T y^{(j)}) \tag{1}$$

where the index q_{ij} denotes the $((i-1)J+j)$-th entry in a vector. Then the

[1] Here we use the VALID padding mehod for simplicity, but other padding methods do not impact our analysis.

image-specific saliency map at the k-th logit is given by

$$\begin{aligned} s_k(x) &= \sum_{i=1}^{N}\sum_{j=1}^{J} h(V_{q_{ij},k})\frac{\partial}{\partial x}\sigma({w^{(i)}}^T y^{(j)}) \\ &= \sum_{j=1}^{J} {D_j}^T \sum_{i=1}^{N} h(V_{q_{ij},k})\frac{\partial}{\partial y^{(j)}}\sigma({w^{(i)}}^T y^{(j)}) \\ &= \sum_{j=1}^{J} {D_j}^T \sum_{i=1}^{N} h(V_{q_{ij},k})\tilde{w}^{(i,j)} \end{aligned} \tag{2}$$

where $h(x) = x$ for plain gradient and $h(x) = \sigma(x)$ for guided backpropagation and $\tilde{w}^{(i,j)} = \begin{cases} w^{(i)} & \text{for } {w^{(i)}}^T y^{(j)} > 0 \\ 0 & \text{otherwise} \end{cases}$.

For an untrained neural network where each entry of both V and W is independent Gaussian distribution with zero mean and variance c^2, i.e. $V_{q_{ij},k}, W_{t,i} \sim \mathcal{N}(0, c^2)\ \forall i \in [N], j \in [J], t \in [p]$. Assuming the number of filters N is sufficiently large (e.g. VGG-16 net usually has $N = 256$), then we have

$$\begin{aligned} s_k(x) &\overset{(a)}{\approx} N\sum_{j=1}^{J} {D_j}^T \mathbb{E}\left[h(V_{q_{ij},k})\tilde{w}^{(i,j)}\right] \\ &\overset{(b)}{=} N\sum_{j=1}^{J} {D_j}^T \mathbb{E}\left[h(V_{q_{ij},k})\right]\mathbb{E}\left[\tilde{w}^{(i,j)}\right] \end{aligned} \tag{3}$$

where (a) follows from the asymptotical approximation of sample mean to the expectation and (b) follows from the fact that $V_{q_{ij},k}$ and $w^{(i)}$ are independent. Next, we need to calculate the two expectations $\mathbb{E}\left[h(V_{q_{ij},k})\right]$ and $\mathbb{E}\left[\tilde{w}^{(i,j)}\right]$, respectively. For plain gradient, $\mathbb{E}\left[h(V_{q_{ij},k})\right] = 0$; for guided backpropogation, $\mathbb{E}\left[h(V_{q_{ij},k})\right] = \sqrt{\frac{2}{\pi}}c$. Also,

$$\begin{aligned} \mathbb{E}\left[\tilde{w}^{(i,j)}\right] &= \int_{{y^{(j)}}^T w>0} w \cdot \frac{2}{(2\pi c^2)^{\frac{p}{2}}} e^{-\frac{w^T w}{2c^2}} dw \\ &\overset{(a)}{=} \int_{z_p>0} U^T z \cdot \frac{2}{(2\pi c^2)^{\frac{p}{2}}} e^{-\frac{z^T z}{2c^2}} |U| dz \\ &\overset{(b)}{=} \sqrt{\frac{2}{\pi}} c \cdot \frac{y^{(j)}}{\|y^{(j)}\|_2} \end{aligned} \tag{4}$$

where z_p is the p-th entry of z, (a) follows from the change of variables $z = Uy^{(j)}$ with U being an unitary matrix satisfying $U \cdot \frac{y^{(j)}}{\|y^{(j)}\|_2} = e_n$, and (b) follows from some algebriac manipulations.

Based on these results, we could explain the behaviors of gradient-based visualizations. On one hand, for plain gradient we have $s_k(x) \approx 0$, which

means there is no information about the input image obtained from the plain saliency map. It explains why we see *random noise* during the plain saliency visualizations in the randomly initialized neural networks.

On the other hand, for guided backpropogation we have

$$\begin{aligned} s_k(x) &\approx \frac{2c^2N}{\pi} \sum_{j=1}^{J} \frac{1}{\|y^{(j)}\|_2} {D_j}^T y^{(j)} \\ &\overset{(b)}{=} \frac{2c^2N}{\pi} \left(\sum_{j=1}^{J} \frac{1}{\|y^{(j)}\|_2} \mathcal{I}_{p_j} \right) x \end{aligned} \tag{5}$$

where (b) follows from the definition $\mathcal{I}_{p_j} \equiv {D_j}^T D_j = \begin{bmatrix} 0_{(j-1)b \times (j-1)b} & & \\ & I_{p \times p} & \\ & & 0 \end{bmatrix} \in \mathcal{R}^{d \times d}$. Ideally, if we assume $\|y^{(j)}\|_2 = C_0$, $\forall j$ (a constant) and ignore the boundary points (note that using the "SAME" padding method to replace the "VALID" one could alleviate the boundary inconsistency to some extent), then $\sum_{j=1}^{J} \mathcal{I}_{p_j} \approx p I_{d \times d}$ and thus we further approximate guided backpropagation as

$$s_k(x) \approx \frac{2c^2Np}{\pi C_0} x \tag{6}$$

Since the scalar term does not matter as we will normalize its value to the range $[0, 1]$ during visualizations, the input image could be *exactly recovered* by a randomly initialized two-layer convnet via guided backpropagation.

How many number of filters N is needed to guarantee a tight estimate in Eq. (3)? From [4], in the high dimensional space, we can approximately have $N = \tilde{O}(\frac{p}{\epsilon^2})$ such that $\| \frac{1}{N} \sum_{i=1}^{N} \tilde{w}^{(i,j)} - \mathbb{E}[\tilde{w}^{(i,j)}] \|_2 < \epsilon$ with high probability, where p denotes the filter size. Here $\tilde{O}(\cdot)$ hides some other constant factors. For example, if setting the filter size to be $3 \times 3 \times 3$ which is commonly used in VGG-16 net, then in order to achieve estimate error 0.1, we might need around 2700 filters. In the next section, our experiments show that $N = 256$ is sufficient to produce a good-looking guided propagation visualization. Despite of being a loose upper bound, it shows the number of filters heavily depends on the dimension of image patches. Since the dimension of image patches is relatively small, we could use a mild number of convolutional filters to recover the input image. That is why we can see clear guided backpropagation visualizations in the most of convnets.

If we replace the convolutional layer by a fully-connected layer in the above example, resulting in a three-layer fully-connected neural network (with no convolution):

$$f_k(x) = \sum_{i=1}^{N_h} V_{i,k} \sigma({w^{(i)}}^T x) \tag{7}$$

where N_h is the number of hidden neurons. Following the same analysis for the three-layer convnet, the fully-connected network can also recover the input image via guided backpropagation by assuming N_h is sufficiently large. But the key difference here is that the dimension of $w^{(i)}$ is d (the dimension of input image) instead, which is very high (e.g. $224 \times 224 \times 3$ for ImageNet). This means we may need enormous hidden neurons here to recover the input image. That is why we cannot see clear guided backprogagation visualizations in the most of fully-connected networks.

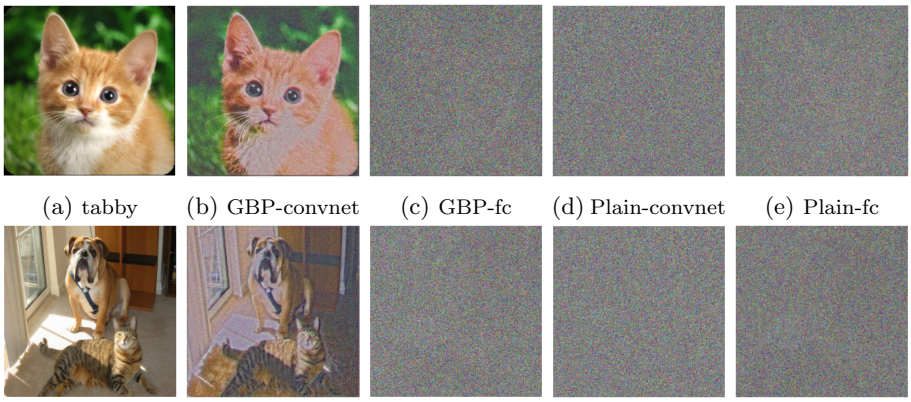
From the above analysis, it is the local connection in the convnets that accounts for most of the human interpretable visualizations. Since the logit label k is arbitrarily chosen in Eq. (1) and (7), our explanation is consistent to our observations in experiments that the gradient-based visualizations are not class-specific.

4 Experiments

To verify our theoretical analysis, we conduct a series of experiments on both three-layer convnets and fully-connected networks with weights being sampled from Gaussian. The input image are randomly chosen from ImageNet of which the dimension is $224 \times 224 \times 3$ unless otherwise stated. For the convnet, by default, the shape of convolutional kernel is $[7, 7, 3, 256]$, which means the filter size is 7×7 and number of filters is $N = 256$. For the fully-connected network, by default, the hidden layer size is $N_h = 4096$.

First, Figure 2 shows both plain saliency map and guided backpropagation visualizations for the convnet and fully-connected network. As we can see, only guided-backpropagation applied in the convnet can produce a clean visualization in our settings. Plain saliency map applied in either the convnet or fully-connected network will result in random noise, which verifies our theoretical analysis. Due to the large dimeansion of input images, the fully-connected network fails to produce any gradient-based visualizations as predicted by our theoretical analysis.

To further verify our theoretical explanations, we conduct two more experiments as follows: For the convnet, we change the number of filters N given other parameters fixed and the results for guided backpropagation are shown in Figure 3. For the fully-connected network, we downsample the input image to be of size $64 \times 64 \times 3$ and increase the hidden layer size N_h, and the results for guided backpropagation are shown in Figure 4. We observe that, as the number of filters N (resp. the hidden layer size N_h) as large as possible, the guided backpropagation visualizations for the convnet (resp. the fully-connected network) become cleaner. These phenomena have been well explained by our theoretical analysis. Note that even by setting $N_h = 100000$, which is definitely unrealistic, the fully-connected network cannot achieve a comparable performance to the convnet with $N = 16$. Therefore, it is the convolutional operation - local connections to be more precise - that really contributes to human interpretability in most of the gradient-based visualizations.



(a) tabby (b) GBP-convnet (c) GBP-fc (d) Plain-convnet (e) Plain-fc

(f) mastiff (g) GBP-convnet (h) GBP-fc (i) Plain-convnet (j) Plain-fc

Figure 2: Guided backpropagation and plain saliency map visualizations for both a three-layer convnet and a three-layer fully-connected network. The first column refers to different input images, and each row corresponds to one specific input image. "GBP" stands for guided backpropagation, and "Plain" stands for plain saliency map. Thus, for example, "GBP-convnet" denotes guided backpropagation for convnet and "Plain-fc" denotes plain saliency map for fully-connected network.

TODO: (1) why plain saliency good for trained vgg ; (2) guided backpropagation captures edges in convnet; (3) adversarial examples

5 Conclusions

In this paper, we gave a theoretical explanation for gradient-based visualizations in deep nerual networks. We revealed that it is the local connections in any convnet that result in human interpretable visualizations no matter the network has been trained or not. Our results suggested that gradient-based visualization mehods convey little information about the value of weights and the importance of pixels to a specific class. Finally, our experiments verified theoretical analysis.

References

[1] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[2] Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., and Dähne, S. (2017). Patternnet and patternlrp–improving the interpretability of neural networks. *arXiv preprint arXiv:1705.05598*.

(a) $N = 8$ (b) $N = 16$ (c) $N = 32$ (d) $N = 64$ (e) $N = 128$

Figure 3: Guided backpropagationvisualizations for a three-layer convnet by varying the number of filters N.

(a) $N_h = 5000$ (b) $N_h = 10000$ (c) $N_h = 40000$ (d) $N_h = 70000$ (e) $N_h = 100000$

Figure 4: Guided backpropagation visualizations for a three-layer fully-connected network by varying the hidden layer size N_h.

[3] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[4] Lugosi, G. and Mendelson, S. (2017). Sub-gaussian estimators of the mean of a random vector. *arXiv preprint arXiv:1702.00482*.

[5] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

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

[7] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[8] Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

[9] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[10] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.