the Interpretability of Deep Neural Networks

Zhang Dinghuai
School of Mathematical Science
Peking University
zhangdinghuai@pku.edu.cn

Abstract

In recent years deep neural networks have achieved outstanding performances, whereas at most of the time the networks can only be seen as "black boxes". In this paper, some attempt in this field are listed, from quantifying and visualizing method to disentangling models.

1 Introduction

The importance of model interpretability has been discussed in [11]. To make a long history short, through interpreting networks we can raise the accuracy and robustness of models, decreasing the reliability on human annotations and march towards unsupervised learning. As a tiny survey, this paper presents a small part of the methods which can be used for representing the interpretability of deep neural networks.

2 Quantifying the interpretability

How to find a proper way of evaluating the interpretability of network is still a chanllenge in the research field. However, only if we have a well-defined method of quantifying the interpretability can we make a judge on the way of interpretability, thus the work of quantifying is really important. Here are some ways which works well.

2.1 Evaluation metric: location stability of a certain filter

In [7] a method of evaluating the location stability was raised. This method is based on the assumption that if a filter f consistently represents a certain object part of objects through many different images, then the distance between location inference of f and the landmark[1] part of objects won't change much. In the above description, the location inference of f means the unit in the feature map with the highest activation value.

Sepecificly, we compute the location stability of filter f like this:

$$d_I(p_k, \hat{\mu}) = \frac{\|p_k - p(\hat{\mu})\|}{\sqrt{\omega^2 + h^2}} \tag{1}$$

$$D_{f,k} = \sqrt{variance_I[d_I(p_k, \hat{\mu})]} \tag{2}$$

In above equations we use $\hat{\mu}$ for the inference of filter f, p_k and $p(\hat{\mu})$for the location of the k-th landmark and inference of f, I for a specific image, w and h for the weight and the height of the picture respectively. Then we use $mean_f mean_{k=1}^{K} D_{f,k}$ to represent our measurement of the location instability of filter f, where K refers to the amount of the landmarks.

[1] According to [9], a landmark is referred to as the central position of a semantic part (a part with an explicit name, e.g. a head, a tail).

2.2 Evaluation metric: part interpretability

This method invented by [1] and promoted by [12] is to measure the fitness between a certain filter and a semantic concept. For a certain filter f and a certain image x, the feature map $A_k(x)$ of it is computed by a CNN, from which we can compute a distribution a_k. Futhermore, we get a threshold T_f which enables the probability $Pr(a_k > T_k) = 0.005$. In order to compare with the input-resolution annotation mask L_c for a specific semantic concept c, we need to scale up $A_k(x)$ to $S_k(x)$ using bilinear interpolation. Then we compute the so-called dataset-wide intersection-over-union score in this way:

$$M_k(x) \equiv S_k(x) \geq T_k \tag{3}$$

$$IoU_{k,c} = \frac{\Sigma|M_k(x) \cap L_c(x)|}{\Sigma|M_k(x) \cup L_c(x)|}, \tag{4}$$

in which $|\cdot|$ means the amount number of a set.

In [1], the authors will link the filter with the semantic concept c once the score is larger than 0.04, *i.e.* $IoU_{k,c} > 0.04$. What's more, we use *unique numbers*, which is the number of unique concepts that are aligned with units in that layer, to describe the interpretability of a layer.

3 Visualization of neural networks: gradient-based method

3.1 Deconvolution methods

Before [6] the visualization of a given high-level feature map cannot go through smoothly. However, in this paper Zeiler and Fergus come up with some new methods such as *Unpooling*[2] and *Rectification* to complete the *deconvolution*, by which we can have a clear look of which part of the input image has the ability to fire a certain neuron in a CNN. The outline of the whole procedure can be seen in Figure1.

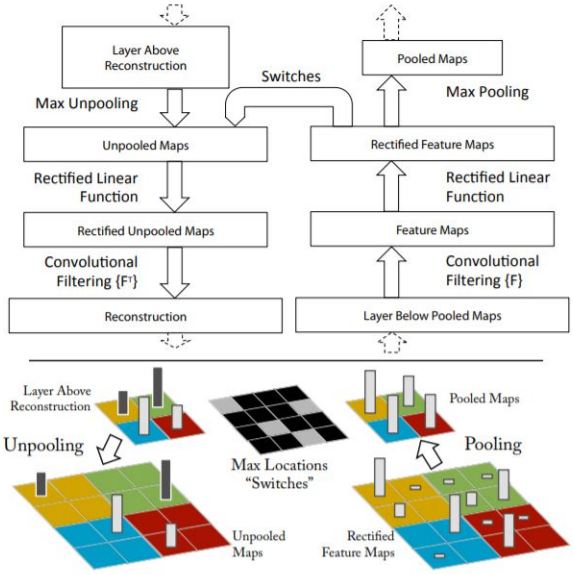

Figure 1: deconvolution

[2]The unpooling uses a variable called *switch* to restore the place of maxima within pooling region. Details can be seen in [6]

In [5] the authors put up with the *guided backpropagation* methods, with which the image generated by deconvolution could become more clear. The main difference between backpropagation, deconvolution and guided backpropagation can be seen in Figure 2.

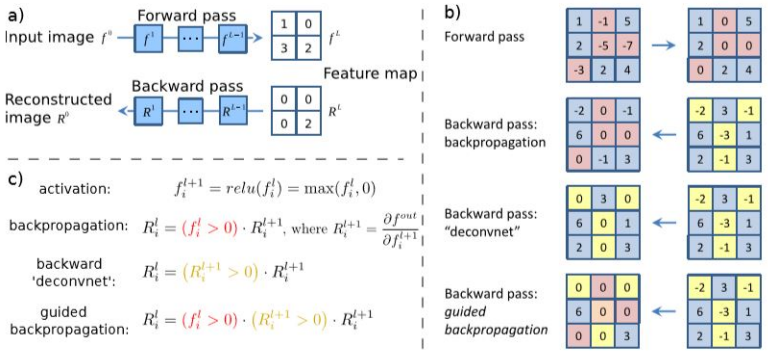

Figure 2: backpropagation, deconvolution, and guided backpropagation

3.2 Visualization by numerous optimization

3.2.1 Gradient ascend

Generally speaking, methods of this kind usually compute the gradient of the score the CNN gives to a certain image and uses this gradient to change the image so that we can generate a new image that gets a higher score, as descirbed in [4].

To be more accurate, a new image can be made by *gradient ascend* like this:

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2, \tag{5}$$

in which I is an image, $S_c(\cdot)$ denotes the score of the class c and λ is a parameter for regularisation. In practice we begin with a image of random noise, then we set gradient of the score of c equal to 1 while others equal to 0 and start the backpropagation. [3] summarizes the progress of this method during the last few years, raising some restiction and promotion of it, such as more ways of regularisation to make the generation picture more interpretable.

3.2.2 Saliency maps

Let I_0 be a given image, c is the class that has the highest score, we can use

$$\omega = \frac{\partial S_c}{\partial I}\Big|_{I_0} \tag{6}$$

to get a vector ω, which can be rearranged to a 3-dimensional matrix H[3]. Then we compute a 2-dimensional matrix M by

$$M_{ij} = max_c \left|\omega_{h(i,j,c)}\right|. \tag{7}$$

Here M is the *saliency map*. In a saliency map of an image, we can easily find out which part of the image contributes most to the final classifcation result as discussed in [4].

However, based on [2], adding a constant shift to input data can bring about the failure of numerous method, *i.e.* saliency method. Maybe more experiments need to be done in this field.

[3] It's obvious that ω has the same shape as I, which is a colorful image with RGB color channels.

4 Disentangling CNNs into interpretable models

4.1 Disentagling CNNs via interpretable CNN

In [9], Zhang *et al.* proposed that each filter in the CNNs should be activated by a certain object part. To accomplete this goal, the authors added an additional loss to each filter in the target conv-layer, encouraging it to encode only one object part appeared only in one object category exclusively as showed in Figure 3.

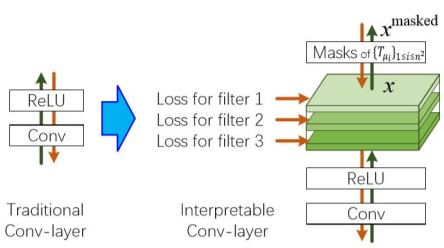

Figure 3: interpretable conv-layer

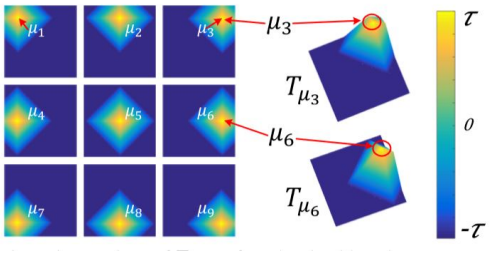

Figure 4: n^2 templates for filter f

In the forward propagation of interpretable CNN, we first make a series of template $\{T_{\mu_1}, T_{\mu_2}, \dots T_{\mu_{n^2}}\}$ as showed in Figure 4 for $n \times n$ feature map x where $T_\mu = (t_{ij}^+)$ and $t_{ij}^+ = \tau \cdot \max(1 - \beta \frac{\|[i,j]-\mu\|_1}{n}, -1)$. Then we compute

$$\hat{\mu} = \arg\max_{[i,j]} x_{ij} \tag{8}$$

and

$$x^{masked} = \max\{x \circ T_{\hat{\mu}}, 0\} \tag{9}$$

where $\circ$ represents the element-wise product.

When doing backpropagation, we add an additional loss term $\lambda \frac{\partial \mathbf{Loss}_f}{\partial x_{ij}}$:

$$\frac{\partial \mathbf{Loss}}{\partial x_{ij}} = \lambda \frac{\partial \mathbf{Loss}_f}{\partial x_{ij}} + \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathbf{L}(\hat{y}_i, y_i^*)}{\partial x_{ij}} \tag{10}$$

$$\begin{aligned} \mathbf{Loss}_f &= -MI(\mathbf{X}; \mathbf{T}) \quad \text{for filter} f \\ &= -\sum_T p(T) \sum_x p(x|T) \log \frac{p(x|T)}{p(x)} \end{aligned} \tag{11}$$

where N is the number of input images. The details and internal meaning of Equation 11 is accessable in [9].

In a word, the Interpretable CNNs can push the filters to detect one distinct object part during training, resulting a more interpretable neural network, not only in the quantification way mentioned in Section 2 but also in the interpretable quantifying way invented in [13].

4.2 Disentangling via desicion tress

The potential desicion modes in CNNs are still somewhat a mystery to human. As a trial in this field, [10] put forward a method mining internal desicion modes of a revised CNN [9] into a desicion tree, *i.e.* given an image, we can use the learned desicion tree to quantitatively explain why this image gets its classification category. This process can be seen intuitively in Figure 5.

As we have modified our model into an Interpretable CNN, now we can assume that each filter in the conv-layer only correspond to one distinct object part, which can be ensured by the design of

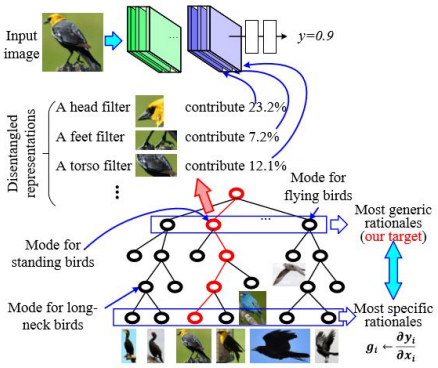

Figure 5: Quantatively explain CNN using desicion tree

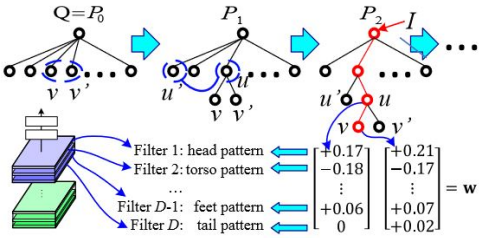

Figure 6: The process of learning a desicion tree

Equation 11. To learn the desicion mode we want, we first initialize a two-layer tree $Q = P_0$. After that, we select two nodes v, v' in the second layer to obtain a new node u for the tree P_n, which we use to replace v, v' in the second layer of the desicion tree, and in the meantime v and v' become the child of u, resulting a new tree P_{n+1}. A visualized version of this process can be seen in Figure 6 and details can be seen in [10].

After having learned a desicion tree, we need to find out a parse tree by which we can explain how an image is classified. For a node u and a given image I, we compute the $cosine$ value between I's gradients and the parameters of u's child nodes, and the child nodes with the maximum computation result becomes part of the parse tree.

From the experimental result we con conclude that when we used a fine-grained model like desicion tree to help do the classification, the prediction error would decrease and the fitness of filter contribution between estimation and reality would increase.

4.3 Disentangling via explanatory graph

The idea of mining internal knowledge of a pre-trained CNN into a graph model can date back to [8]. Simaliarily, [7] put forward a method using explanatory graph 7 to find out the relationship between part patterns and convolutional filters.

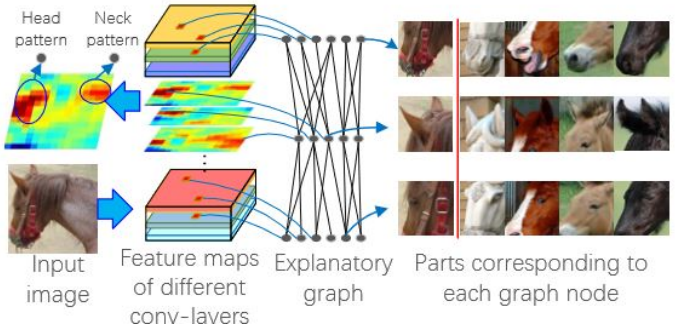

Figure 7: Explanatory graph and its relationship with CNN

Here are more infomation about the graph: A explanatory graph has similar multiple layer construction as CNNs, specificly, each layer of the explanatory correspond to a layer of the CNN. Every layer of explanatory graph is consist of many nodes, as every layer of CNN is consist of many filters. Each node is correspond to a certain part pattern, while one certain filter has relationship with several part patterns, thus a filter has connection to several nodes. The relationship between part patterns and nodes is relatively robust, which means it won't change much through different input images. What's more ,between different layers there are edges linking nodes encoding latent spatial relationship in adjacent layers.

Like what we do while learning desicion trees, we learn a explanatory graph from a pre-trained CNN using its feature maps. The graph can be seen as some sort of dictionary, for each input image, we can "look for" its corresponding part patterns (or, nodes) that determine the result of classification.

5 Conclusion and prospect

As we can see, many methods have been invented to give the neural network a proper interpretability. However, we have to acknowledge that human are still at a very young stage of understanding internal logics of networks, especially with the emergence of adversarial examples which interpretable models cannot explain. A sharp problem is : if these interpretable models are getting the right way, then why there exist adversarial examples that can be easily distinguished by human beings? Perhaps there are some issues in these models that can't be ignored, waiting for us to discover.

Acknowledgments

Thanks to my handsome supervisor Zhu Zhanxing for offering inspiring advice in the field of interpretability of neural networks. Besides, during the process of my wrinting this tiny survey, my classmates Lu Yiping (who is the *fatest* person I've ever seen in the campus), Li Tongyu and Jia Zeyu give me some suggestion on using LaTeX, since it's the first time that I use this powerful writing tool, thus I shall extend my thanks to them.

References

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *CoRR*, abs/1704.05796, 2017.

[2] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (Un)reliability of saliency methods. *NIPS workshop on Explaining and Visualizing Deep Learning*, 2017.

[3] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[4] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[5] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[6] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[7] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting CNN knowledge via an explanatory graph. *CoRR*, abs/1708.01785, 2017.

[8] Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. Multi-shot mining semantic part concepts in cnns. *CoRR*, abs/1611.04246, 2016.

[9] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. *CoRR*, abs/1710.00935, 2017.

[10] Quanshi Zhang, Yu Yang, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnns via decision trees. *CoRR*, abs/1802.00121, 2018.

[11] Quanshi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *CoRR*, abs/1802.00614, 2018.

[12] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *CoRR*, abs/1711.05611, 2017.

[13] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014.