GDR-TAL Paper review: All that glitters is not gold

Romain XU-DARME [1,2]

[1]CEA-LIST/DILS/LSL [2]Laboratoire d'Informatique de Grenoble (LIG)

Sanity checks for saliency maps, NeurIPS 2018

Explanations can be manipulated and geometry is to blame, NeurIPS 2019

Why do we need to visualize important input features?

- Debugging
- Convincing
- Learning(?)

(a) Husky classified as wolf

(b) Explanation

Feature importance

Notations:

- Classifier $\mathcal{M} : \mathbb{R}^d \rightarrow \mathbb{R}^c$
- Input $x \in \mathbb{R}^d = (x_1, \ldots, x_d)$, prediction $y = \mathcal{M}(x)$.

How does a change on one feature of x impact output of $\mathcal{M}$?

Feature importance

Notations:

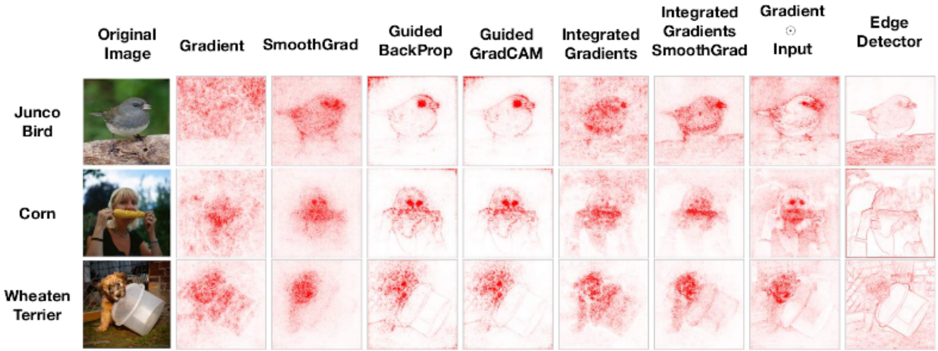
- Classifier $\mathcal{M} : \mathbb{R}^d \rightarrow \mathbb{R}^c$
- Input $x \in \mathbb{R}^d = (x_1, \ldots, x_d)$, prediction $y = \mathcal{M}(x)$.

How does a change on one feature of x impact output of $\mathcal{M}$?

A large ecosystem:

1. Model-agnostic (black-box methods)
   - SHAP [1], *LIME* [2], Anchors [3], RISE [4], ...
2. Model-specific (white-box methods)
   - Gradients $\frac{\delta \mathcal{M}}{\delta x_i}(x)$: Backprop, SmoothGrad [5], Grad-CAM [6], ...
   - Image $x \odot$ Gradients: Integrated gradients [7], LRP [8], DeepLIFT [9], ...
   - Deconvolution: DeConvNet [10], Guided-backpropagation [11], ...

Examples of feature visualization



Sanity Checks for Saliency Maps [12]

Sanity Checks for Saliency Maps

Julius Adebayo*, Justin Gilmer♯, Michael Muelly♯, Ian Goodfellow♯, Moritz Hardt♯†, Been Kim♯
juliusad@mit.edu, {gilmer,muelly,goodfellow,mrtz,beenkim}@google.com
♯Google Brain
†University of California Berkeley

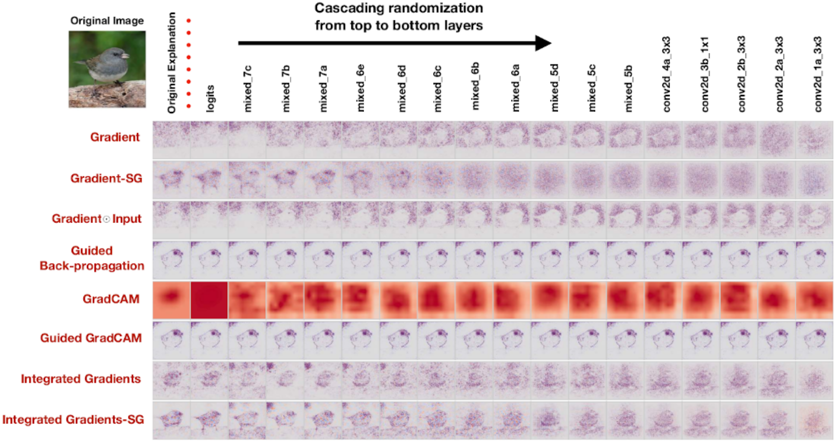
Sanity Checks for Saliency Maps

1. Feature importance should depend on the parameters of the model $\mathcal{M}$
2. Feature importance should depend on the label of the prediction

Sanity Checks for Saliency Maps

1. Feature importance should depend on the parameters of the model $\mathcal{M}$
   - Progressively replace the weights of the trained model $\mathcal{M}$ with random values
   - See impact on visualization
2. Feature importance should depend on the label of the prediction

Does the visualization depends on the parameters of the model?



Does the visualization depends on the parameters of the model?

- Methods based on Guided-backpropagation seem insensitive to parameter randomization
- Methods based on $x \odot \frac{\delta M}{\delta x}$ seem to retain some information about the input
- Methods based solely on $\frac{\delta M}{\delta x}$ seem OK (with some weird artifacts)

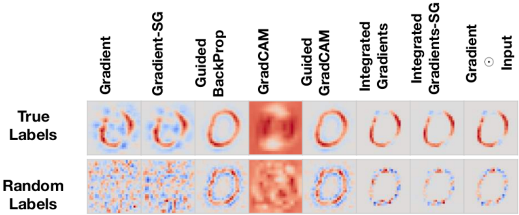
Sanity Checks for Saliency Maps

1. Feature importance should depend on the parameters of the model $\mathcal{M}$
2. Feature importance should depend on the label of the prediction

Sanity Checks for Saliency Maps

1. Feature importance should depend on the parameters of the model $\mathcal{M}$
2. Feature importance should depend on the label of the prediction
   - Shuffle training set labels and retrain model
   - See impact on visualization

Does the visualization depends on the label of the prediction?



- Methods based on Guided-backprop seem insensitive to label randomization.
- So do methods based on Image ⊙ grads, with the caveat that MNIST contain very sparse images
- Again, methods based solely on gradients seem OK

Does the visualization depends on the label of the prediction?

An alternative (not in the paper): keeping the original trained model, what happens if we ask guided-backprop for the visualization of a random label (say the least likely categories in a classification task)?

Conclusion on Sanity Checks for Saliency Maps

- According to the paper, some widely used visualization methods seem to exhibit the same characteristics as edge detectors
  - Regarding guided-backprop methods, there may exist a bias in the way "random" weights are initialized and something about the natural expressiveness of modern Deep-CNN architectures.
- But once the doubt is here, we may wonder: how deep goes the rabbit hole?

Conclusion on Sanity Checks for Saliency Maps

- According to the paper, some widely used visualization methods seem to exhibit the same characteristics as edge detectors
  - Regarding guided-backprop methods, there may exist a bias in the way "random" weights are initialized and something about the natural expressiveness of modern Deep-CNN architectures.
- But once the doubt is here, we may wonder: how deep goes the rabbit hole?

Spoiler alert: it goes pretty deep...

Using adversarial setting to generate any explanation [13]

Explanations can be manipulated and geometry is to blame

Ann-Kathrin Dombrowski[1], Maximilian Alber[5], Christopher J. Anders[1], Marcel Ackermann[2], Klaus-Robert Müller[1,3,4], Pan Kessel[1]

[1]Machine Learning Group, Technische Universität Berlin, Germany
[2]Department of Video Coding & Analytics, Fraunhofer Heinrich-Hertz-Institute, Berlin, Germany
[3]Max-Planck-Institut für Informatik, Saarbrücken, Germany
[4]Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea
[5]Charité Berlin, Berlin, Germany

Using adversarial setting to hide model bias [14]

Fooling Neural Network Interpretations via Adversarial Model Manipulation

Juyeon Heo[1,*], **Sunghwan Joo**[1,*], and **Taesup Moon**[1,2]
[1]Department of Electrical and Computer Engineering, [2]Department of Artificial Intelligence
Sungkyunkwan University, Suwon, Korea, 16419
heojuyeon12@gmail.com, {shjoo840, tsmoon}@skku.edu

Conclusion

- It is not because a visualization method provide a pretty saliency map matching the object that it is necessarily accurate ($\approx$ confirmation bias)
- Systemic fooling of visualization methods by modifying the model itself can open up the door to developers hiding some failures of their model under the carpet (e.g. a model with a non-ethical bias), with(currently) no way of detecting such failures.

References I

[1] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *NIPS*, 2017.

[2] M. T. Ribeiro, S. Singh, and C. Guestrin, ““why should i trust you?”: Explaining the predictions of any classifier,” in *NAACL 2016*, 2016.

[3] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in *AAAI*, 2018.

[4] V. Petsiuk, A. Das, and K. Saenko, “Rise: Randomized input sampling for explanation of black-box models,” in *BMVC*, 2018.

[5] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, “Smoothgrad: removing noise by adding noise,” *ArXiv*, vol. abs/1706.03825, 2017.

References II

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

[7] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 3319–3328, JMLR.org, 2017.

[8] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *ICANN*, 2016.

References III

[9] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153, PMLR, 06–11 Aug 2017.

[10] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[11] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," *CoRR*, vol. abs/1412.6806, 2015.

[12] J. Adebayo, J. Gilmer, M. Muelly, I. J. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *NeurIPS*, 2018.

References IV

[13] A.-K. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," in *NeurIPS*, 2019.

[14] J. Heo, S. Joo, and T. Moon, "Fooling neural network interpretations via adversarial model manipulation," in *NeurIPS*, 2019.