Essay on Understanding Deep Networks

Georgios M. Moschovis, MSc Machine Learning student, 19970325-7536
KTH Royal Institute of Technology, geomos@kth.se

1. Introduction

One of the most exciting technological aspects nowadays is Machine Learning's mind-blowing potential in transforming the world we live in, especially due to its exciting resurgence through Deep Learning. However, as machine learning models are becoming more complex, there is a noticeable trade-off between *accuracy* and simplicity or *interpretability*[1] and plenty of cutting-edge research papers have been published in top-tier conferences related to this tension. In this essay we will focus on some recent developments in this topic that give promising results towards the ultimate goal of a general Artificial Intelligence (AI).

The core idea behind *interpretability* is to specify how a particular model makes decisions, how certain it is about them, whether and in what extent it can be trusted or how it may be corrected especially in mission-critical applications[2]. In this context, *interpretability* could be beneficial to particular applications of Deep Learning, for instance in *Diagnostic Captioning*[3] models, where it is essential for an AI system to explain its generated captions, in the sense of including all important findings and not referring to the wrong findings and reasoning its decisions to ensure trust with the medical community. Robotics is another field where *interpretability* plays a crucial role, for instance in autonomous driving agents, as well as in collaboration between humans and robots under the scope of Human-Robot Interaction (HRI)[2].

Furthermore, the principal goals of *interpretable* AI boil down to identifying failure modes that enables developers to further improve their systems, establishing the appropriate trust and confidence to users, including the direction of fairness, which requires eliminating the biases to be incorporated in a model, as well as "*machine teaching* a human" on improving their own decisions[4].*Interpretable* AI models are either white-box or black-box and cover either a specific type of networks or some wider range of those.

2. Methods and representative works

GradCAM[4] is a popular *interpretability* method applied to a variety of CNN architectures that weighs feature activations in different pixel regions within an image with the average gradient of the class scores. After these gradients are computed through global average pooling, they are passed to a $ReLU$(i) that intensifies pixels contributing towards increasing the target class activation scores. Compared to its predecessor CAM[5] that is a special case of GradCAM, it is not restricted to image classification CNNs without fully-connected layers but is also providing saliency maps that are both class discriminative and high resolution, thus may be considered as more interpretable and widely applicable.

These desirable properties in the generated heatmaps could be further enhanced when fusing Guided Backpropagation with the activations through element-wise multiplication leading to more detailed coarse heatmaps. Furthermore, using the negative gradients results in removing concepts occurring in regions that decrease the network's confidence, further improving the quality of saliency maps. What is more, GradCAM can provide useful explanations for failure modes through its visualizations, produce competitive segmentation results compared to CAM when trained in low-data regime, as well as relatively low classification and localization errors on ILSVRC-15 validation set. It is also invariant to adversarial attacks as in several experiments even if it was perturbated to predict a wrong class with very high accuracy, the explanation for this prediction was the background, as well as for any other irrelevant class, while for the actual

(i) REctified Linear Units activation function is: $ReLU(x) = \max\{x,0\}$.

classes it provided accurate importance maps.

In addition, a plethora of experiments involving humans to evaluate class discrimination, trust, faithfulness, and interpretability of GradCAM, with the latter pair to be considered negatively correlated, demonstrated its interpretability and faithfulness capabilities. When considering a biased dataset, it tends to make biased predictions as well, however, this bias is eliminated, and its explanations become more accurate when using a balanced dataset instead. Moreover, regarding image captioning tasks, GradCAM accurately provides evidence within an image for the individual words included in the COCO[6] captions[(ii)] and when evaluated on bounding boxes for regions of interest produced by DenseCap[7], it accurately localizes each bounding box even though not trained with the respective annotations.

Although GradCAM demonstrates impressive results, it is restricted to CNN architectures. One a more general approach is **RISE**[8] that measures pixels' importance by applying element wise multiplications the original input with a sampled random binary mask to reduce their intensities to zero and only preserve the most important among them. The respective importance scores are computed by a black-box framework and importance maps are computed by probing the network being ran on masked inputs and considering the weighted average of the masks. This approach however is prone to adversarial effects. A possible improvement to reduce storage requirements is to sample smaller masks, then upsample them using bilinear interpolation and crop additional regions, to get a saliency map.

Moreover, an important family of explanation methods includes those based in the concept of **attribution**. Assuming F: $\mathbb{R}^n \to [0,1]$ is the learnt predictive function, we define as attribution a vector $\alpha \in \mathbb{R}^n$, where each element α_i refers to the contribution of the i^{th} element of the input x_i ($i \leq n$) to the prediction $F(x)$[9]. This vector α is defined with respect to some baseline x', which ideally indicates the absence of some cause. In Natural Language Processing settings (NLP) the all-zero embedding is a good baseline, while in vision applications the black image is used, in order to model lack of importance in the limit and assign blame of the prediction to particular features' subset in the input. In addition, the association between the baseline input x' and the original input $x = h_x(x')$ is termed as a mapping fuction[10].

In order to demonstrate the importance of attribution methods, we should briefly mention some axioms and properties that they satisfy, as well as their importance in developing *interpretable* AI models. *Sensitivity* requires that if for any pair of baseline input x' and original input x, which although differ in one feature i have different predictions, then the differing feature should be given a non-zero attribution. In mathematical notation $\exists i$ s.t. $x_i \neq x_i'$, $F(x_i) \neq F(x_i') \Rightarrow \varphi_i \neq 0$. One also important property is *implementation invariance*, which requires for two models that are functionally equivalent –i.e. produce the same predictions for every input, to also generate the same attribution; thus $\exists F_1, F_2$ s.t. $F_1(x) = F_2(x)\ \forall x \Rightarrow \varphi^{(1)} = \varphi^{(2)}$. Although super critical, those axioms are usually violated in practice, for instance gradients violate sensitivity i.e. in cases of functions being flattened when reaching a plateau and "discrete gradients"[9] used by methods such as Deep Taylor Decomposition[11] break implementation invariance.

One example of an attribution method designed on the principle of respecting these axioms is **integrated attribution**[9]. This approach relies on integrated gradients being accumulated at all points along the straight-line path from the baseline input x' to the original input x, defined as the "path integral of the gradients along it"[9]. Even more interestingly, apart from *sensitivity* and *implementation invariance*, this method satisfies a property called *completeness* stating that these attributions sum to the difference model prediction at the original input x, $F(x)$ and the prediction at the baseline input x', $F(x')$, thus $\sum \partial F/\partial x = F(x) - F(x')$ that is extremely useful for debugging as well as explaining the decided attributions. Furthermore, removing the constraint

(ii) Evaluation was performed using *The Pointing Game* metric.

of the straight-line path and taking into account all possible paths γ: $[0,1] \to \mathbb{R}^n$, this method is generalized to a whole family of *path methods*[9]. Apart from the aforementioned properties and axioms, *path methods* additionally satisfy *linearity* i.e. using two models F_1, F_2 to form another one $F(x) = \alpha F_1(x) + \beta F_2(x) \Rightarrow \varphi = \alpha\varphi^{(1)} + \beta\varphi^{(2)}$, $\alpha, \beta \in \mathbb{R}$, as well as *preserve symmetry* for any baseline inputs x' and original inputs x that have identical values for symmetric variables, i.e. x_1, x_2 s.t. $F(x_1,x_2) = F(x_2,x_1)$, also receive identical attributions.

Integrated attribution model and *path methods* in general demonstrate interesting qualitative results in NLP, performing machine translation or predicting the type of answer some question is seeking, as well as in biology, attributing the cause of a disease in the responsible parts of a lesion or predicting its severity, but also whether a molecule is active against a protein or enzyme through biochemical reactions. In the latter experiment, a molecular graph convolution architecture is being used in combination with *path methods*.

Last but not least, **local attribution methods**[10] are extremely important as well. Building on the notation of mapping function local methods will try to ensure $g(z') \approx f(h_x(z'))$ when $z' \approx x'$. We distinguish attribute feature attribution methods where generative model $g(z')$ computes a sum of all features attributions effects, denoted as $g(z') = \sum_i \varphi_i \zeta_i \approx f(h_x(z'))$, $\zeta = \{1, z'\}$, $z' \in \{0,1\}^M$, M, $\varphi_i \in \mathbb{R}\ \forall i$. Among those, LIME[12] uses a local linear explanation model, DeepLIFT transforms binary baselines into values in the original input space[13,14], LRP[15] is equivalent but restricted to the all-zero baseline, while Classic Shapley Value Estimation includes *Sharpley regression values*[16], Sharpley sampling values[17] and Quantitative input influence[18]. Among those Shapley methods, the first computes attributions as differences on the model predictions on all possible feature subsets $S \subseteq F_{\text{all}} \backslash \{i\}$ and the full features' space F_{all}, while the latter could be considered as sampling approximations of the former.

This class of methods also allows us to distinguish several interesting properties, such as *local accuracy* that requires the explanation model g to match at least the original network prediction $f(x)$ for the simplistic baseline x', $g(x')$, *missingness* that requires features not occurring in the input x to have no impact on the decision and thus zero attributions, $x_i' = 0 \Rightarrow \varphi_i = 0$; but also *consistency* between models, suppose F_1, F_2 satisfying the following derivation for an input z': $F_2(z') = F_2(z' \mid z_i' = 0) \geq F_1(z') = F_1(z' \mid z_i' = 0)\ \forall\ z' \in \{0, 1\}^M \Rightarrow \varphi_i(F_2, x) \geq \varphi_i(F_1, x)$. These properties require a unique formulation of an explanation model g and a set of values adhering to all of them, SHAP values. These are estimated using either model-agnostic or model-type-specific approximation methods such as LIME or DeepSHAP respectively. The former uses a so-called Sharpley kernel while the latter assumes feature independence and linearity, considers parts of the network, computes SHAP values in each and then aggregates them for the full network through DeepLIFT's multipliers during back-propagation. Its approximations are impressively accurate despite the simplifying assumptions.

3. Comparison and conclusion

As mentioned earlier, different types of networks serve the goals of *Interpretability*, which are mainly identified as white-box or black-box. Each of these model classes might satisfy better the need for explaining different target networks' decisions, white-box methods usually provide higher-quality importance maps but are limited to particular network architectures. An example demonstrated in this essay is GradCAM that although provides impressive saliency maps is limited to CNN architectures. Black box explanation models are typically more generic, in the sense that they cater for a larger variety of models, however results are typically less inspiring. What is more, a theoretical concept of attribution values can yield meaningful properties in the design of those explanation models. Lately, transformers have been involved in the design of such models[19], delivering promising results.

4. References

1. Gilpin et al., “Explaining Explanations: An Overview of Interpretability of Machine Learning”, *IEEE 5th International Conference on Data Science and Advanced Analytics*, 2018
2. Azizpour, “Session 12 – Understanding Deep Networks”, DD2412/FDD3412 October 2021
3. Pavlopoulos et al., “Diagnostic Captioning: A Survey”, 2021, ArXiv, abs/2101.07299.
4. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization", *ICCV* 2017
5. Zhou et al., “Learning Deep Features for Discriminative Localization”, *CVPR* 2016.
6. Chen et al., “Microsoft COCO Captions: Data Collection and Evaluation Server”, 2015
7. Johnson et al., “DenseCap: Fully Convolutional Localization Networks for Dense Captioning”, *CVPR* 2016.
8. Petsiuk et al., "RISE: Randomized Input Sampling for Explanation of Black-box Models", *BMVC* 2018
9. Sundararajan et al., "Axiomatic Attribution for Deep Networks", *ICML* 2017
10. Lundberg et al., "A Unified Approach to Interpreting Model Predictions", NIPS 2017
11. Montavon et al., “Explaining nonlinear classification decisions with deep Taylor decomposition”
12. Ribeiro et al., “Why should i trust you?: Explaining the predictions of any classifier”. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *ACM*, 2016.
13. Shrikumar et al., “Learning Important Features Through Propagating Activation Differences”, 2017, abs/ArXiv 1704.02685
14. Shrikumar et al., “Not Just a Black Box: Learning Important Features Through Propagating Activation Differences”, 2016, abs/ArXiv 1605.01713
15. Bach et al. “On pixel-wise explanations for non-linear classifier decisions by layerwise relevance propagation”, *PloS One* 10.7, 2015, e0130140.
16. Lipovetsky et al., “Analysis of regression in game theory approach”, *Applied Stochastic Models in Business and Industry* 17.4, 2001.
17. Štrumbelj et al., “Explaining prediction models and individual predictions with feature contributions”, *Knowledge and information systems* 41.3, 2014.
18. Datta et al., “Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems”, *Security and Privacy (SP), 2016 IEEE Symposium on IEEE* 2016
19. Chefer et al., "Transformer interpretability beyond attention visualization", *CVPR* 2021

5. Self assessment for bonus

Last but not least, considering the amount of details that I describe per paper that is a result of long studying sessions of these works, I would to nominate myself for bonus. Several of the aforementioned details relate to sharing commonalities and/or making comparisons with other publications (I have listed more than 15 references). After the essay deadline you are welcome to visit the github repository referenced hereunder, where I upload my *.pdf* annotations both on the “Selected Papers for Essays” and other related works on *Interpretability* as well as my study notes in form of slides.

georgmosh/dla_assg2: Essay assignment on Deep Networks Interpretability (github.com)

Stockholm, October 19, 2021