UNIVERSITY OF CAMBRIDGE

Axiomatic Attribution for Deep Networks

L193 - Explainable Artificial Intelligence

Background

Research Question: How do we attribute the prediction of a DNN to its input features?

Some Applications:

- Object Recognition: which pixels are most responsible for the resulting label?
- Question Classification: which words signal what type of answer a question expects?
- Machine Translation: which input tokens do output tokens mostly correspond to?

Informative for both end-users and developers

Background

How do we formally define the attribution of an input feature?

Let F represent a DNN and x an input:

$$F: R^n \rightarrow [0, 1]$$
$$x = (x_1, \ldots, x_n) \in R^n$$

Then attribution from a baseline x' is $\mathrm{a_F(x, x')} = (a_1, \ldots, a_n) \in R^n$ where a_i is the contribution of x_i to $F(x)$

An intuitive choice of baseline should be an input where $F(x') \approx 0$ and convey an absence of signal (e.g., black image)

Motivation

Challenges:

1. Existing methods are hard to evaluate empirically – how to delineate model vs. attribution method misbehavior is not clear
2. Existing methods tend to require modifications to the network and/or are costly to compute

Objectives:

1. Define axioms that, when satisfied, addresses the issue of evaluation
2. Introduce a method that satisfy the axioms and is easy to compute/implement

Axioms

1. Sensitivity
    a) For all pairs of x and x' that differ by one feature which results in different predictions, then the differing feature should have $a_i > 0$
    b) If F does not mathematically depend on an x_i then $a_i = 0$
2. Implementation Invariance
    a) Attributions should be the same for two functionally equivalent DNNs
3. Completeness
    a) Attributions should add up to $F(x) - F(x')$
    b) This is a stronger version of Sensitivity (a)
4. Linearity
    a) If $G = i \times F_1 + j \times F_2$ then $a_G(x, x') = i \times a_{F_1}(x, x') + j \times a_{F_2}(x, x')$

Axioms

Examples of violations of axioms:

1. Sensitivity
   - Gradients – e.g., $F(x) = 1 - ReLU(1 - x)$ fails with $x' = 0$ and $x = 2$
   - Causes focus on irrelevant features
2. Implementation Invariance
   - Methods that use “discrete” gradients (finite difference approx.) such as DeepLift and Layer-wise Relevance Propagation (LRP) as chain rule does not generally hold
   - Undesirable that attributions can arbitrarily change even if the “black box” outputs are the same

Integrated Gradients

Motivation: Gradients satisfy Implementation Invariance, methods like DeepLift and LRP satisfy Sensitivity, can we combine?

Yes – integrate gradients on the line connecting x and x' in R^n. Formally:

$$IntegratedGrads_i(x) := (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

Integrated Gradients

Integrated Gradients satisfy:

- Completeness by the fundamental theorem of calculus for path integrals

$$\sum_{i}^{n} IntegratedGrads_i(x) = F(x) - F(x')$$

- Sensitivity (a) by Completeness
- Sensitivity (b) as if $F(x)$ does not depend on x_i then $\frac{\partial F}{\partial x_i} = 0$ for all values of x_i
- Implementation Invariance by only considering instantaneous gradients
- Linearity as differentiation obeys linearity

$$\int_{\alpha=0}^{1} \frac{\partial[a \times F_1(\ldots) + b \times F_2(\ldots)]}{\partial x_i} d\alpha = a \int_{\alpha=0}^{1} \frac{\partial F_1(\ldots)}{\partial x_i} d\alpha + b \int_{\alpha=0}^{1} \frac{\partial F_2(\ldots)}{\partial x_i} d\alpha$$

Integrated Gradients

Integrated Gradients is a specific case of Path Methods:

$$PathIntegratedGrads_i^{\gamma}(x) \coloneqq \int_{\alpha=0}^{1} \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha$$

where γ is a smooth function from $[0, 1] \rightarrow R^n$ specifying a path from x' to x. Integrated Gradients is the case where $\gamma(\alpha) = x' + \alpha \times (x - x')$.

All path methods satisfy Completeness, Sensitivity, Linearity, and Implementation Invariance. Integrated Gradients is the unique case that also preserves symmetry.

Integrated Gradients

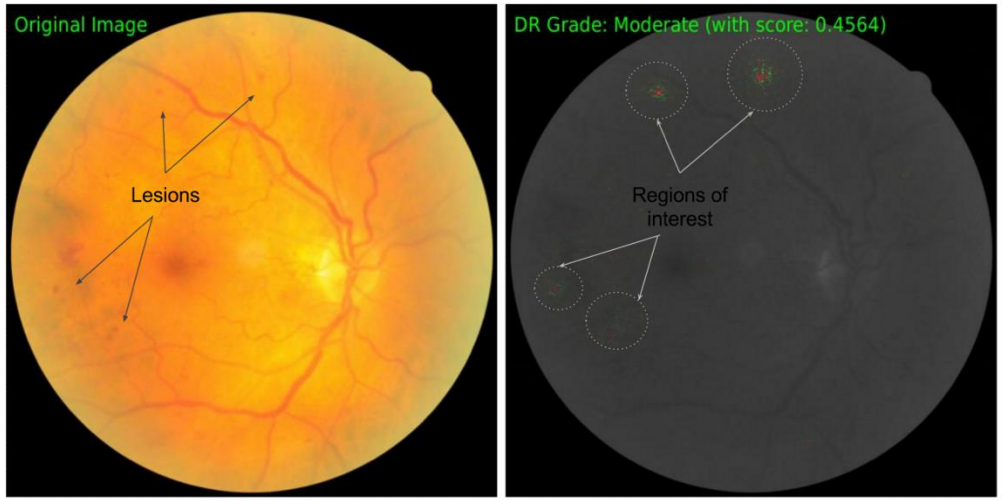
Efficient calculation via Riemann approximation:

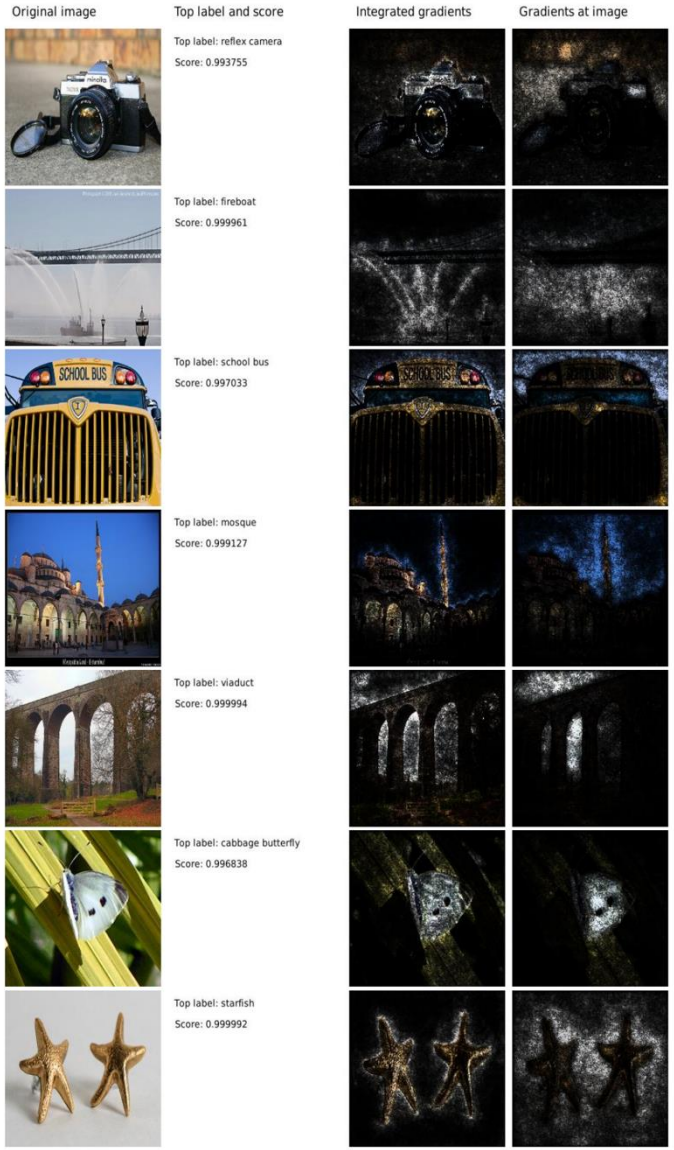
$$IntegratedGrads_i(x) \approx (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}$$

Gradients can be easily obtained from most deep learning frameworks (e.g., `tf.gradients` in TensorFlow or `torch.autograd.grad` in PyTorch).

Furthermore, m can be easily tuned simply by comparing the Riemann sum to $F(x) - F(x')$ due to the Completeness axiom being satisfied.

Results


Original Image
Lesions
DR Grade: Moderate (with score: 0.4564)
Regions of interest


Original image
Top label and score
Integrated gradients
Gradients at image
Top label: reflex camera
Score: 0.993755
Top label: fireboat
Score: 0.999961
Top label: school bus
Score: 0.997033
Top label: mosque
Score: 0.999127
Top label: viaduct
Score: 0.999994
Top label: cabbage butterfly
Score: 0.996838
Top label: starfish
Score: 0.999992

Future Work

Predictive Power:

- Integrated Gradients only provides attribution evaluated at a specific point
- It does not address how attribution values change, the only way to find the attribution for another point is to evaluate Integrated Gradients at the new point.

Input Feature Interactions:

- Attributions are specific to each feature; we do not know the importance of combinations of input features

Baselines:

- Newly released model architectures and different modalities might not have an obvious neutral baseline, finding them is an ongoing process