Critique of Right for the Right Reasons: Training Neural Networks to be Interpretable, Robust, and Consistent with Expert Knowledge

William Jardee willjardee@gmail.com
Physics
Montana State University
Bozeman, MT 59715, USA

Editor:

1. Introduction

2. Related Works

2.1 Explainable Models

2.2 Ensemble and Adversarial Methods

2.3 Human Centered Interpretability Measures

2.4 Interpretable Representations

3. Right for the Right Reasons Algorithm

The first novel idea introduced by the paper is a loss function that discourages the impact of gradients in regions declared by a mask matrix, A. This loss function is presented in the context of neural networks. It is pointed out that all of the functions provided are differentiable and thus can be minimized with gradient methods. The first proposed iteration of the process is

$$\mathcal{L}(\theta, \mathbf{X}, y, A) = \underbrace{\sum_{n=1}^{N}\sum_{k=1}^{K} -y_{nk}\,\log(\hat{y}_{nk})}_{\text{Right Answers (Cross-Entropy)}} + \underbrace{\lambda_1 \sum_{n=1}^{N}\sum_{d=1}^{D}\left[A_{nd}\frac{\partial}{\partial x_{nd}}\sum_{k=1}^{K}\log(\hat{y}_{nk})\right]^2}_{\text{Right Reasons}} + \underbrace{\lambda_2 \sum_i \theta_i^2}_{\text{Regularizer}}. \quad \text{(Equation 1)}$$

Where θ are the input parameters, $\mathbf{X}$ is the input vector, and y are the target values. λ_1, λ_2, and A are provided hyperparameters. The authors point out that λ_1 should be chosen such that the "Right Answers" and "Right Reasons" are of the same magnitude; they go on later to analyze the impact of varying the parameter. A is the annotation matrix that masks unwanted gradients that can be provided via domain knowledge or recursively defined. The cross-entropy and regularizer terms are standard and the new "Right Reasons" accounts for gradients in directions that are not preferred.

The authors introduce the "Find-Another-Explanation" model where multiple neural nets are learned sequentially:

$$\begin{aligned} A_0 &= 0 & \theta_0, &= \arg\max_\theta \mathcal{L}(\theta, \mathbf{X}, y, A_0) \\ A_1 &= M_c[f_x|\theta_0], & \theta_1 &= \arg\max_\theta \mathcal{L}(\theta, \mathbf{X}, y, A_1) \\ A_2 &= M_c[f_x|\theta_1] \cup A_1, & \theta_2 &= \arg\max_\theta \mathcal{L}(\theta, \mathbf{X}, y, A_2) \\ &\dots & &\dots \end{aligned}$$

where M_c is a binary mask that activates for features that reached a critical threshold, c, in the last stage. When the annotation matrix is not changed between subsequent runs, or when λ_1 must be tuned high in comparison to previous runs, then this model has spanned the whole of the viable model space and the set of resulting models can be passed to a domain expert to select the proper reason.

In Equation (1), specifically when the annotation matrix is not designed by an expert, each subsequent run is unique from following runs. This means that if a previous run had a mixture of reasons, it is likely that all of them will be disregarded for future runs. This can be problematic if important and unimportant reasons show up in the same annotation matrix. To account for this, it is proposed that reasons will be locally independent. Formally, this is stated as

$$f(x) = f(x_{g_{\max}}) \quad \forall\, \epsilon' < \epsilon$$

such that $x_{g_{\max}} = \arg\max g(x')$, $x' \in N_{\epsilon'}(x)$, and $N_{\epsilon'} = \mathcal{B}_\epsilon(x) \cap \Omega_x$, where $\mathcal{B}_\epsilon(x)$ defines a hypersphere in the feature space Ω_x with radius ϵ.

Replacing the annotation matrix reliant loss function in Equation (1) with the idea of local independence provides

$$\mathcal{L}(\{\theta_m\}) = \underbrace{\sum_m \mathbb{E}_{p(x,y)}\left[-\log(p(y|f(x;\theta_m))\right]}_{\text{Predictive Term (Cross-Entropy)}} + \underbrace{\lambda \sum_{l \neq m} \mathbf{IndepErr}\left(f(\cdot;\theta_m), f(\cdot;\theta_l)\right)}_{\text{Diversity Measurement}} \tag{Equation 2}$$

$$\approx \underbrace{\sum_m \mathbb{E}_{p(x,y)}\left[-\log(p(y|f(x;\theta_m))\right]}_{\text{Predictive Term (Cross-Entropy)}} + \underbrace{\lambda \sum_{l \neq m} \mathbf{CosIndepErr}\left(f(\cdot;\theta_m), f(\cdot;\theta_l)\right)}_{\text{Diversity Measurement}} \tag{Equation 3}$$

where

$$\begin{aligned} \mathbf{IndepErr}(f,g) = \mathbb{E}\left[\left(f(x_{g_{\max}}) - f(x)\right)^2\right] \approx \left(\epsilon\, \boldsymbol{\nabla} f(x) \cdot \boldsymbol{\nabla} g(x)\right)^2 \rightarrow \left(\frac{\boldsymbol{\nabla} f(x) \cdot \boldsymbol{\nabla} g(x)}{|f(x)|_2 |g(x)|_2}\right)^2 \\ \equiv \cos^2(\boldsymbol{\nabla} f(x), \boldsymbol{\nabla} g(x)) \equiv \mathbf{CosIndepErr}(f,g). \end{aligned}$$

The latter conclusion where **IndepErr** is approximately the cosine error is achieved by two derivations. The first is a logical argument from the first order Taylor expansion that then

gets to the cosine similarity after normalizing the result. The second is by considering the covariance between the change in two functions and minimizes the two after doing a Gaussian approximation. It should be noted that the final algorithm only has one hyperparameter that needs to be tuned, and clear guidance is given on how to pick it.

Equation (2) naturally flows from the idea of Equation (1) when the idea of recursively generating the annotation matrix is shifted to choosing the A that maximizes local independence. Choosing the optimal $x_{g_{\max}}$ is very difficult and can either be closely approximated with adversarial methods, as will be expanded on later, or using a linear approximation when $\epsilon \to 0$. The latter is what allows the function to be written as Equation (3).

4. Algorithm Experiments

5. Application of Algorithm in Adversarial Context

6. Interpretability with Human's Study

7. Conclusion