Classifier-Agnostic Crossing of Decision Boundary under the Image Manifold

Anonymous Authors[1]

Abstract

1. Introduction

Understanding the behavior of complex machine learning predictors is important for debugging and for improving their performance and robustness. However, performance alone is not enough – even if a classifier has 100% accuracy for a given class, it could still fail to understand that particular class or concept. For example, if all images of a polar bear exhibit a white background, the classifier can learn to make the decision based on the background pixels. Therefore, methods that *interpret* what the classifier learned can help flag hidden biases in the data (**?**), evaluate the fairness of the model (**?**), and build trust in the system [XX].

Saliency methods (**????**) are popular tools for interpreting predictors by highlighting the most important features in an image yielding the given classification. While saliency methods have proven useful for a variety of applications [XX], some of them were shown to be agnostic to the predictor model and hence cannot help with debugging (**?**).

Non-salient methods of interpretability have also been developed, that rely on local occlusions or perturbations to the input (**??**). Yet another class of methods have focused on counterfactual explanations (**??**), trying to show in what way does the input need to change in order to be assigned a different class. However, one disadvantage of these methods is that such perturbations are not guaranteed to generate realistic looking images, which diminishes their usefulness for end users (e.g. clinicians) (**?**).

Recently, some methods (**???**) have used generative adversarial netoworks (GANs) in order to explore the image manifold and use it to interpret black-box classifier results. However, the design is sub-optimal as it requires re-training the GAN every time the black-box classifier is modified, which is something that can happen often, especially when debugging through progressive refinement. This is especially difficult for state-of-the-art (**??**) generators that are complex to train and require large ammounts of computational power. While (**?**) employ variational-autoencoders which are easier to train, they do not yield realistic looking images.

We propose a method for visualising a black-box classifier by generating realistic images that cross the classifier's decision boundary. More precisely, we assume a fixed, pre-trained generator G that can generate realistic images. This makes our work applicable not only for uncovering biases in the classifier, but also for debugging and improving its performance through progressive refinement. Since our method does not require training the generator, we can also use pre-trained off-the-shelf generators available online.

***I'm thinking**: Probably we should add active learning as an objective as well. The idea is that once we identify a generated image that is wrongly classified, we can add it along with the correct label to the training dataset. This augmented dataset would hopefully improve performance. If we do this, we can then build a system that generates many paths from "promising" starting points, and augment the dataset with manually labeled images that are close to the boundary.*

2. Related work

3. Method

We assume a black box classifier $f : X \rightarrow R$, an initial input image $x \in X$, and a target class C. Without loss of generality, we assume the classifier is binary and that $f(x) \in [0, 1]$ is the probability of assigning x to target class C. Moreover, compared to (**?**), we also assume a fixed, pre-trained generator $G : Z \rightarrow X$, generating realistic images. We aim to find a trajectory of T synthetic images $\mathbf{x} = (x_1, ..., x_T)$ corresponding to latent points $\mathbf{z} = (z_1, ..., z_T)$, $z_i \in Z$ such that:

- **Monotonicity**: the trajectory monotonically converges towards the target class C: $f(z_t + 1) > f(z_t)$
- **Identity preservation**: latent points z_t are as close as possible to the embedding of the source image $G^{-1}(x)$. For example in computer vision or medical imaging, this helps preserve the identity of the person studied, ensuring only minimal changes are made in order to change the label of the point to target class C.

4. Option 1: Gradient descent along the classifier softmax

Without loss of generality, assume $f(x) = 0$. Start with initial point $z_0 = G^{-1}(x)$ and generate points z_{t+1} as follows:

$z_{t+1} = z_t + \eta \nabla f(G(z_t))$

5. Option 2: Reguarised optimisation

We define a "step" loss on classifier f, ensuring that the target value corresponding to z_t is $\frac{t}{T}$:

$$L_f(\mathbf{z}) = \sum_{t=1}^{T-1} |f(G(z_{t+1})) - \frac{t}{T}| \tag{1}$$

The above loss ensures monotonicity in the assignment of f to class C as we progress along the image manifold. We further define an identity preservation loss, e.g. to ensure that the anatomy of the subject provided in image x is preserved:

$$L_{id}(\mathbf{z}) = \sum_{t=1}^{T-1} |G^{-1}(x) - G(z_t)| \tag{2}$$

The overall loss is a weighted sum of each loss component:

$$L(\mathbf{z}, \lambda) = L_f(\mathbf{z}) + \lambda L_{id}(\mathbf{z}) \tag{3}$$

Parameters $\mathbf{z}$, λ need to be optimised.

6. Results

7. Discussion