An Adversarial Approach for Explaining the Predictions of Deep Neural Networks

Arash Rahnama
Modzy
arash.rahnama@modzy.com

Andrew Tseng
Modzy
andrew.tseng@modzy.com

Abstract

Machine learning models have been successfully applied to a wide range of applications including computer vision, natural language processing, and speech recognition. A successful implementation of these models however, usually relies on deep neural networks (DNNs) which are treated as opaque black-box systems due to their incomprehensible complexity and intricate internal mechanism. In this work, we present a novel algorithm for explaining the predictions of a DNN using adversarial machine learning. Our approach identifies the relative importance of input features in relation to the predictions based on the behavior of an adversarial attack on the DNN. Our algorithm has the advantage of being fast, consistent, and easy to implement and interpret. We present our detailed analysis that demonstrates how the behavior of an adversarial attack, given a DNN and a task, stays consistent for any input test data point proving the generality of our approach. Our analysis enables us to produce consistent and efficient explanations. We illustrate the effectiveness of our approach by conducting experiments using a variety of DNNs, tasks, and datasets. Finally, we compare our work with other well-known techniques in the current literature.

1. Introduction

Explaining the outcomes of complex machine learning models is a prerequisite for establishing trust between the machines and users. As humans increasingly rely on DNNs to process large amounts of data and make decisions, it is crucial to develop solutions that can interpret the predictions of DNNs in a user-friendly manner. Explaining the outcomes of a model can help reduce bias and contribute to improvements in model design, performance, and accountability by providing beneficial insights into how models behave [10]. Consequently, the field of explainable artificial intelligence systems, XAI, has gained traction in recent years, where researchers from different disciplines have come together to define, design and evaluate explainable systems [33, 7, 25]. The majority of current explainability algorithms for DNNs produce an explanation for a single input-output pair: an input data point fed into the DNN and the respective prediction made by the DNN. The algorithm usually finds the most important features in the input contributing the most to the model's predictions and selects those as explanations for the model's behavior [2]. The majority of these algorithms find the important features using either a *perturbation-based* approach or a *saliency-based* approach [20]. The saliency-based approaches rely on gradients of the outputs in relation to the inputs to find the important features [31, 29]. Perturbation-based methods on the other hand apply small local changes to the input, track the changes in the output, and find and rank the important input features [27, 1].

One main problem with current state-of-the-art explainability tools is their reliance on a large set of hyper-parameters. This leads to local instability of explanations and can negatively affect the user's experience [2]. An explainability algorithm should satisfy 3 properties: 1- It has to produce human-understandable explanations, 2- It has to be locally consistent and efficient, 3- It should be user-friendly, easy to apply and quick in providing explanations. In this work, we propose a new algorithm, explanations via adversarial attacks, which satisfies these 3 important properties and more. We call our method **A**dversarial **E**xplanations for **A**rtificial **I**ntelligence systems or AXAI [1]. AXAI inherits from the nature of adversarial attacks to automatically find and select important features affecting the model's prediction to produce explanations. The idea behind our work comes from the natural behavior of adversarial attacks. The attacks tend to manipulate important features in the input to deceive a DNN. The logic is simple, rather than trying to build a model that learns to explain the DNN's behavior, why don't we utilize the nature of attacks to learn this behavior? One who knows how to fool a model, certainly knows what the model may be thinking. Another

[1] Code will be readily available.

benefit of our approach is that certain attacks, such as the Projected Gradient Descent (PGD) method [21], are fast, efficient, and consistent in their adversarial behavior. Our work further aims to solve at least 2 problems: 1- Provide fast explanations without a need for model training, 2- Reduce the need for selecting a large set of hyper-parameters to produce consistent results.

Obviously, one needs to first show how adversarial attacks link to explainbility, i.e., how an attack can point to the important features in the input and how one can filter out the unimportant ones to produce explanations. Further, one needs to show how an adversary behaves similarly in its approach across models, tasks and datasets so that the explanations are consistent, stable, and applicable to a large group of models. Here, we present a novel algorithm for explaining the DNN's predictions in multiple domains including text, audio and image. In particular, this paper makes the following contributions:

- We show that given an ℓ_2 PGD attack and a trained DNN, the distribution of attack magnitudes vs. frequency across all unseen test inputs follows a beta distribution, regardless of the task and dataset. We also show that these distributions are symmetric and the differences between their means, medians, and quantiles are not statistically significant.

- We show that the most important input features, i.e., features with the largest effect on the model's predictions, can be found using a consistent rule across different DNN architectures, datasets, and tasks. This rule leverages the properties of the distributions explained above.

- We propose a novel algorithm for explaining the outcomes of DNNs and provide a detailed analysis of our algorithm's performance for different DNN architectures, datasets and tasks.

- We benchmark our algorithm against methods such as LIME and SHAP [20, 27] and show that our algorithm performs faster while producing similar or better explainability results.

2. Related Work

One of the popular explainability solutions called LIME [27] assumes that DNNs are linear locally. LIME trains weighted linear models on the top of the DNN for perturbed samples around a target input to produce explanations. The computational bottleneck in LIME is caused by the training part where a selected number of perturbed samples are sent through the DNN for learning the explanation. Certain combination of LIME's hyper-parameters can produce unstable results [2]. DeepLIFT produces explanations by modeling the slope of gradient changes of output with respect to the input [30]. Grad-CAM is a saliency-based method that uses the gradients of the input at the final convolutional layer to produce coarse localization maps pointing to important regions in the input [29]. The majority of approaches based on sensitivity maps fail to produce explanations that only rely on important features. Creators of DeepLIFT associate this lack of stability to the behavior of activation functions such as ReLU. [32] proposed Smooth Grad which uses gradients and Gaussian based de-noising methods to produce stable explanations. The authors of the paper mention that large outlier values in the gradient maps produced by gradient differentiation may cause instability. In our algorithm, we overcome the problem of instability by utilizing the density of attacks, which are created iteratively on segments. Some other important works in this area are given in [34, 14, 39, 4, 5, 9, 18].

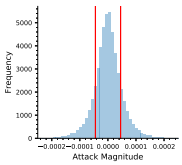
DNNs are vulnerable to subtle adversarial perturbations applied to their input. The basic idea behind most adversarial attacks revolves around solving a maximization problem with a constraint that keeps the distance between the original input and adversarial input small, so that the adversarial input, while capable of fooling the DNN, is not perceptually recognizable by humans. The connection between model interpretation and attacks has recently gravitated the interest of researchers. [13] and [35] showed that one benefit of adversarial examples is that they reveal useful insights into the salient features of input data and their effects on DNNs' predictions. Our solution relies on the nature of adversarial attacks to select and produce important and explainable features given a specific input and DNN. Our work puts more emphasis on model interpretability, where we make use of the information obtained from an adversarial attack on a DNN to de-noise the sensitivity maps and produce stable explanations. We de-noise the gradient map by utilizing the iterative nature of the PGD attack and by considering only a minimum number of highly influential gradients that contribute the most to the predictions. We use the density of gradients in a number of segments to remove the noise that was not filtered out in the previous steps and produce human-interpretable explanations.

3. Main Results

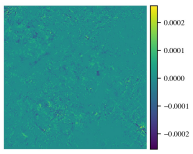
The core idea behind our approach, AXAI, is to utilize the knowledge gained from an adversarial attack on a DNN and an input, to find the important features in the input in order to produce good explanations. This is done by mapping "carefully filtered attacked inputs" onto predefined segments and filtering out the unimportant features. This will be discussed in more detail in later sections. First let's look at an example in Fig. 1 to see how our approach works. Given an image classification DNN, the ℓ_2 adversarial attack changes the pixels in the entire image, as seen in

Fig. 1c. The reason for this is simple: each pixel value is changed by the adversary so that the accumulated loss value can increase enough to fool the DNN. Fig. 1b shows the distribution of the attack on this image. The x-axis represents the magnitude of the pixel changes and the y-axis represents the number of pixels given each value on x-axis. AXAI maps the strongly attacked pixels to the image segments of the original image and filters out the segments with highest density of attacked pixels which meet certain criteria to produce explanations. Fig. 1c shows the value changes for the important attacked pixels. As we will show, the important features used for explanations are located at specific sections in the tails of the distribution given in Fig. 1b. These are the pixels that directly affect the classification decision made by the model. We use QuickShift [37] for segmenting the input image (Fig. 1d). Fig. 1e shows the explanation produced by our algorithm.

(a) Original image



(b) Attack Magnitude vs. Freq.



(c) Adversarial changes in pixels

(d) Image segments

(e) Explanation

Figure 1: A simple example depicting the steps taken in AXAI to produce explanations.

Algorithm 1 details the steps taken by AXAI to produce an explanation E for the output of a selected model f. Suppose that input X is segmented into p groups using a segmentation method and that the attack magnitudes for the input X and DNN f are obtained. Let X_{diff} be the difference between the original X and adversarial X'. We filter out the low intensity attack magnitudes X_{diff} and create a Boolean array X_{difft}, where values larger than a threshold, are only set to True. Let Su be the set of unique segments, $Su = \{Su_1, \ldots, Su_p\}$. Next, we map the filtered attack X_{difft} to the segments Su, and create a new list of filtered attack groups, $Su_x = \{Su_{x_1}, \ldots, Su_{x_p}\}$. The mapping function, *Map* in Algorithm 1, simply stacks the filtered attacks on the segments and groups the filtered attack X_{difft} based on the segments. Finally, the attack density of each unique segment can be written as $Su_d = \{\frac{card(Su_{x_1})}{card(Su_1)}, \ldots, \frac{card(Su_{x_p})}{card(Su_p)}\}$ (*Calculate_density* in Algorithm 1). We then extract the indices j's of the top K maximum values in Su_d (*TopK_indices* in Algorithm 1), and produce $Su(j)$ as explanation E for the input X. In next sections, we explain each step in details.

Algorithm 1 AXAI

Require: Model f, input X

1: $X' \leftarrow Attack(f, X)$ ▷ i.e. PGD attack
2: $X_{diff} = x' - x$ ▷ The attack magnitudes
3: $X_{difft} \leftarrow Threshold(X_{diff})$ ▷ Filtered attack magnitudes
4: $Su \leftarrow Segment(X))$
5: $Su_x \leftarrow Map(X_{difft}, Su)$ ▷ Group attack magnitudes based on segmentation
6: $Su_d \leftarrow Calculate\_density(Su_x)$ ▷ Calculate attacks per segment
7: **return** $Su(TopK\_indices(Su_d))$

3.1. White-box adversarial attacks

Adversary can attack a DNN by adding engineered noise to the input to increase the associated loss value, if it has some prior knowledge of the DNN including the weights and biases. AXAI utilizes Projected Gradient Descend (PGD) attack [21], although any ℓ_2 adversarial attack can replace PGD in our algorithm (Appendix B). However, PGD provides specific benefits such as stability and gradient smoothness that other attacks do not. PGD can be thought of as an iterative version of ℓ_2 Fast Gradient Method (FGM) attack [12], where in each iteration, the adversarial changes are clipped into an ℓ_2 ball of some ϵ value. PGD is generally considered a strong stable attack and is defined as,

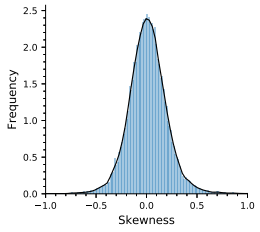
$$x^{t+1} = \sqcap_{x+S}(x^t + \epsilon \nabla_x L(\Theta, x, y)), \quad (1)$$

where for t iterations, x and y are the inputs and outputs, and Θ are the weights and biases.

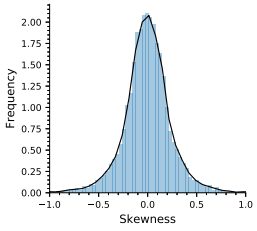
3.2. Statistical analysis of attack magnitudes vs. frequency distributions

Here, we briefly report our statistical analysis of attack magnitudes vs. frequency distributions for a fixed DNN, dataset and an adversarial attack. We can show that the distributions are similar in their "shapes," "means," "mean ranks," "medians," and "quantiles," and follow a Beta distribution with specific parameters. Given that there is no significant difference in the distributions, we can provide a universal threshold using quantiles which separates the important features from the rest to produce explanations.

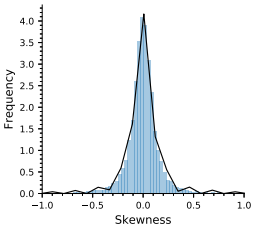
We can measure the symmetricity of distributions using the Fisher-Pearson coefficient of skewness. We present



(a) PGD, AlexNet, CIFAR10



(b) PGD, VGG16, CIFAR100



(c) PGD, ResNet34, ImageNet

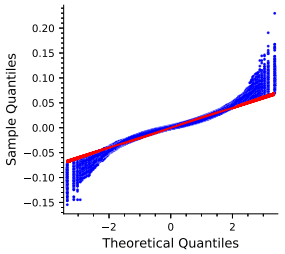
Figure 2: The Fisher-Pearson coefficient of attack magnitudes vs. frequency distributions.

the results for AlexNet on CIFAR10 [15], VGG16 on CIFAR100 [17] and ResNet34 on ImageNet [8]. The Fisher-Pearson coefficients of the attack magnitudes vs. frequency distributions for all cases are shown in Fig. 2. It is seen that the skewness of all distributions falls within the $[-0.5, 0.5]$ range showing strong evidence that they are approximately symmetric [6]. Only 0.9% of CIFAR10, 3.3% of CIFAR100 and 1.9% of ImageNet test datasets lie outside of $[-0.5, 0.5]$ range.

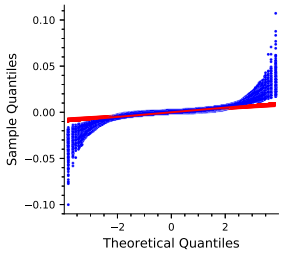
Quantile-Quantile (Q-Q) plot allows us to understand how the quantiles of a distribution deviate from a specified theoretical distribution. The theoretical distribution selected is the normal distribution. The x-axis and y-axis represent the quantile values of the theoretical and sample distributions, respectively. While it is unlikely to have identical distributions that perfectly match, one can look at different parts of the Q-Q plot to distinguish between the similar and dissimilar locations in the distributions. Fig. 3 shows the Q-Q plots for random subsets of ImageNet and CIFAR10 test datasets each containing 1000 images. It is seen that the distributions follow a fairly straight line in the middle portion of the curve, while deviating at the upper and lower parts. This provides some evidence supporting the hypothesis that distributions follow a 'near-normal' distribution with heavier tails.

We perform the two-sample location t-test and Mann-Whitney U test to determine if there is a significant difference between two groups where the null hypothesis is the equality of the means. Carrying out pair t-tests on all samples allows us to be conservative in confirming the mean similarity of the distributions. A sample here is defined as the attack magnitudes vs. frequency distribution for a data point in the test adversarial dataset created by the PGD attack on a DNN trained on the training dataset. The results reported in Table 1 indicate no significant difference between the means. Further, the Mann-Whitney U test results indicate that all pairs are similar to each other on the mean ranks. Under the assumption of two distributions having similar shapes, one could further state that Mann-Whitney test can be considered as a test of medians [22]. Since, we have shown that the shapes are similar, we can conclude that there are no significant difference between the medians of the distributions. Further details in addition to the results for the ANOVA test are given in Appendix C.

Next, to show consistency across distributions for a given model, dataset and attack, we estimate the values of quantiles, means and medians. We do this by estimating the statistics of the distributions and constructing confidences intervals. For each experiment, we estimate the mean, median, 15th, 25th, 75th and 85th quantiles of each attack mag-



(a) PGD, AlexNet, CIFAR10



(b) PGD, ResNet34, ImageNet

Figure 3: The Q-Q plot of sample distributions vs. theoretical normal distribution (mean=0, std=1).

	t-test (CIFAR10)	Mann-Whitney (CIFAR10)	t-test (ImageNet)	Mann-Whitney (ImageNet)
p-value	0.70	0.58	0.64	0.55

Table 1: p-values for the mean similarity statistical tests at significance level 0.05.

	AlexNet, CIFAR10, PGD	VGG16, CIFAR100, PGD	ResNet34, ImageNet, PGD
15th Quantile	$(-1.807e-02, -1.805e-02)$	$(-1.419e-02, -1.414e-02)$	$(-1.785e-03, -1.777e-03)$
25th Quantile	$(-1.145e-02, -1.071e-02)$	$(-8.153e-03, -8.110e-03)$	$(-1.015e-03, -1.101e-03)$
Mean	$(1.775e-05, 2.295e-05)$	$(-6.850e-06, -3.624e-06)$	$(-1.090e-07, -6.000e-08)$
Median	$(2.115e-06, 1.127e-05)$	$(-2.842e-06, 4.467e-06)$	$(-2.155e-07, -9.381e-08)$
75th Quantile	$(1.071e-02, 1.073e-02)$	$(8.102e-03, 8.146e-03)$	$(1.011e-03, 1.016e-03)$
85th Quantile	$(1.809e-02, 1.812e-02)$	$(1.413e-02, 1.418e-02)$	$(1.777e-03, 1.785e-03)$

Table 2: Estimations for mean, median, 15th , 25th, 75th and 85th quantiles at 95% confidence level.

	AlexNet, CIFAR10, PGD	VGG16, CIFAR100, PGD	ResNet34, ImageNet, PGD
p	$(1.124e+01, 1.132e+01)$	$(2.129e+01, 2.171e+01)$	$(1.306e+02, 1.329e+02)$
q	$(1.136e+01, 1.145e+01)$	$(2.124e+01, 2.164e+01)$	$(1.303e+02, 1.326e+02)$

Table 3: Statistical estimations for parameters of beta distribution at 95% confidence level.

	CIFAR10, AlexNet	ImageNet, ResNet34		CIFAR10, AlexNet	ImageNet, ResNet34
Attack Percentile			Attack Percentile		
$15\% - 85\%$	0.78	0.88	$0\% - 15\%\&85\% - 100\%$	0.16	0.07
$10\% - 90\%$	0.26	0.79	$0\% - 10\%\&90\% - 100\%$	0.26	0.13
$5\% - 95\%$	0.50	0.63	$0\% - 5\%\&95\% - 100\%$	0.45	0.25
$1\% - 99\%$	0.07	0.12	$0\% - 1\%\&99\% - 100\%$	0.92	0.80

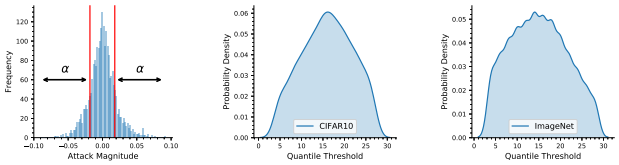
Table 4: Adversarial test accuracy where only features within a certain percentile of the attack magnitudes vs. frequency distributions are attacked (PGD with 20 Iterations).

nitude vs. frequency distribution for the entire test dataset. The statistical confidence interval estimations at confidence level of 95% are reported in Table 2. Our results show that the confidence intervals have narrow ranges and the estimations are consistent. The estimates for the 15th, 25th, 75th and 85th quantiles indicate a strong symmetricity with respect to the origin in all cases. This matches the results of the skewness test in Fig. 2. Another observation is that the confidence interval of the mean and medians are pretty narrow, supporting the results of the t-tests and Mann-Whitney U test. Finally, we can show with high confidence that the distributions consistently follow a beta distribution. The beta distribution is a family of distributions defined by two positive shape parameters, denoted by p and q. The estimated p and q of the beta distribution are reported in Table 3. Further technical details on our analyses presented in this section, in addition to further experiments with audio and text input types, are provided in Appendix C.

3.3. Quantile selection for the explanations

Our algorithm produces explanations that rely only on the features in the input that have the largest effect on the predictions. While the majority of the input is attacked, our belief is that only important features are strongly attacked. We show how one can select the boundary threshold between "explainable features" and the rest based on attack magnitudes. We demonstrate this with 2 experiments: 1) AlexNet trained on CIFAR10, 2) ResNet34 trained on ImageNet, both attacked by PGD with 20 iterations. In each case, we select the successfully attacked inputs from the adversarial test dataset, i.e., the inputs that fool the DNN. We then only re-attack specific features of the original clean inputs within the $[0\%, \alpha\%]$ and $[(100-\alpha)\%, 100\%]$ percentile of the distributions, where α is the percentage threshold. The re-attacking process starts from $\alpha = 0$, where none of the input features are attacked, and then we gradually increase the value of α until the attack successfully changes the prediction, and then we save the value of α (Fig. 4a). We repeat this for every input. The probability density distribution of α's are given in Fig. 4b and Fig. 4c with an estimated mean of $\alpha = 15$.

Further, we report the test accuracies of the DNNs on the adversarial test datasets that are created based on different attack percentiles. Given an attack percentile range, the adversarial test dataset consists of adversarial test inputs which are created by attacking only portions of the input features that lie withing a specific percentile range of the attack magnitudes vs. frequency distributions similar to above. This allows us to understand how the features lying in the middle area, tails and outliers of the distributions affect the DNN's predictions. Our findings are reported in Table 4. Our results show that the majority of the input features including those within the first two standard deviations and the outliers of the distributions do not have a strong effect on the predictions. A smaller portion of the input features which are also those attacked with the highest intensity, i.e., within the $[0\%, 15\%]$ and $[85\%, 100\%]$ percentiles of the distributions have the largest effect on the DNN's predictions, confirming our hypothesis. We see the same trend across different DNNs and datasets (Appendix C).



(a) (b) (c)

Figure 4: Visualization of the re-attacking process where only portions of inputs lying outside the red lines are attacked ($[0\%, \alpha\%]$, $[(100-\alpha)\%, 100\%]$) (b) AlexNet, CIFAR10 (c) ResNet34, ImageNet

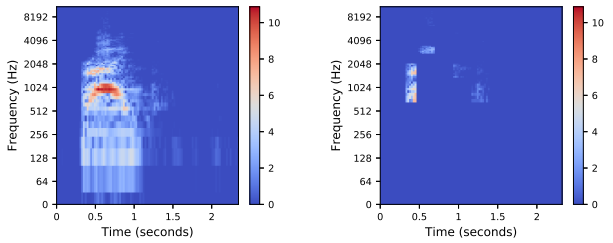
4. Experiment Results

Earlier, we provided a sample explanation created by AXAI for an image classifier. Appendix E contains more experiments for image classification and object detection DNNs. Further, Appendix E contains an ablation study and an interesting comparison between explanations produced by a non-robust model and an adversarially robust model. Here, we provide sample explanations produced by our algorithm for speech recognition and language-based tasks.

4.1. Explaining a speech recognition model

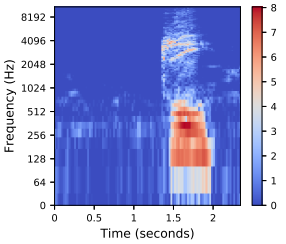
The Speech Commands Dataset [38] is an audio dataset of short spoken words. Here, we have converted the audio files to spectrograms and used them to train a LeNet model to identify "speech commands." We have created time-frequency segments by dividing the spectrogram into time-frequency grids similar to [23]. The x-axis and y-axis indicate the time-scale and log-scale frequency of the spectograms respectively, and the color bar indicates the magnitude. The spectrogram of the first word "Right" and its explanation are shown in Fig. 5a and Fig. 5b. The explanation shows that the first and last character in the spoken word "Right" stand out as important features ($[0.4s, 0.6s]$ and

$[1.0s, 1.2s]$ intervals). This is reasonable because "Five" is the neighboring class of "Right" in the dataset (Appendix D) and "Right" and "Five" differ in the pronunciation of "r" and "f" and "t" and "v." The second example is for the word "Three" (Fig. 5c and Fig. 5d). The produced explanation indicates the importance of "Thr" ($[1.4s, 1.7s]$ interval). This is reasonable because "Three" and its neighbor "Tree" differ in the letter "h" in "Thr," and this difference is learned by the model during training to identify the two words correctly. More examples are shown in 8. Details on this experiment are given in Appendix E.

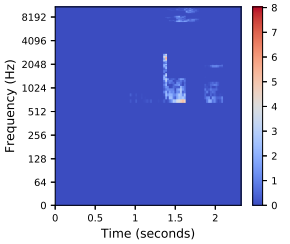


(a) "Right"

(b) Explanation



(c) "Three"

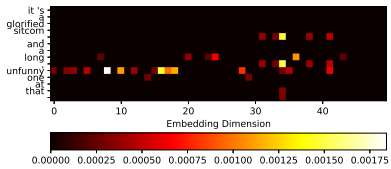


(d) Explanation

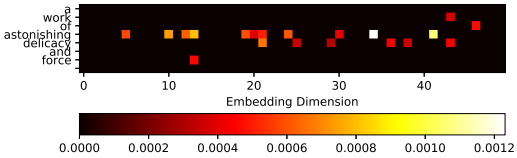
Figure 5: The AXAI explanations for the LeNet speech recognition model.

4.2. Explaining a text classification model

The Sentence Polarity Dataset [26] is a collection of movie-review documents labeled with respect to their overall sentiment polarity. Here, we will look at a negative and positive example (Fig. 6a and Fig. 6b) where the rows are the word tokens in the sentence, and the columns are the embedding dimensions. The NLP model used in our experiment is taken from [16] and trained on the dataset. As part of the pre-processing, the words in the dataset are tokenized and mapped to an embedding matrix. [19] mentions that the saliency map of an NLP model can be visualized using the embedding layer similar to saliency maps used for image-based models. Consequently, one can apply our algorithm to NLP models in a similar manner, i.e., we can utilize the first order derivative of the loss with respect to the word embedding. This technique is similar to what was used in [24]. The first example, "it's a glorified sitcom, and a long, unfunny one at that." is classified as a negative review by the



(a) Text example 1



(b) Text example 2

Figure 6: The AXAI explanations for the sentence classification model.

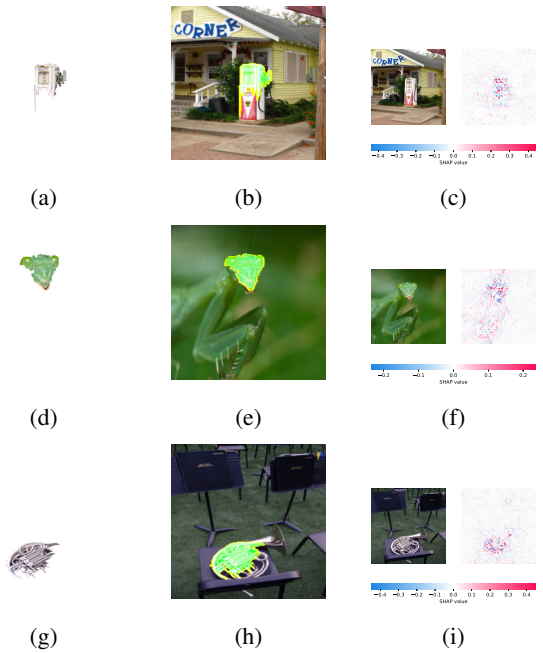
model. Fig. 6a shows that the word "unfunny" is strongly highlighted as the main explanation for this prediction. For the positive example "a work of astonishing delicacy and force," it is seen that the word "astonishing" has the most significant influence on model's prediction. More examples are shown in Fig. 9.

4.3. Benchmark tests

We test our algorithm against LIME and SHAP (Gradient Explainer). It is important to note that SHAP subsumes a number of prior approaches and provides a fair baseline. To show the consistency of our approach, we present visualizations for 3 cases: 1) AlexNet, CIFAR10, 2) VGG16, CIFAR100, 3) ResNet34, ImageNet using the 3 explainability tools and provide more experiments in Appendix F. The algorithms produce similar explanations where AXAI has fewer tune-able parameters and performs faster. LIME fails to produce good explanations for low-resolution CIFAR10 images. In Appendix F, we provide examples showing that AXAI outperforms LIME for low-resolution inputs. We benchmark the running-time performance of AXAI, LIME and SHAP for ResNet34 trained on ImageNet on a single CPU (Intel Core i5-7360U) and single GPU (Tesla V100-SXM2) on the entire test dataset. The results are given in Table. 5. LIME is the slowest to produce explanations. This is because LIME needs to forward propagate the perturbed inputs through the DNN several times. SHAP is also slower to generate the results in comparison to AXAI. LIME works better on a GPU. AXAI maintains its relative performance on the CPU and GPU. This is because the segmentation step which mainly uses the CPU is the main computational bottleneck for the algorithms (Appendix A). A few comparisons between AXAI, LIME, and SHAP are shown in Fig. 7.

	Single CPU (Intel Core i5-7360U)	Single GPU (Tesla V100-SXM2)
LIME	105s	5.8s
SHAP (Gradient Explainer)	35s	3.8s
AXAI (PGD with 20 iters)	6.6s	1.7s

Table 5: Benchmark running-time experiments.

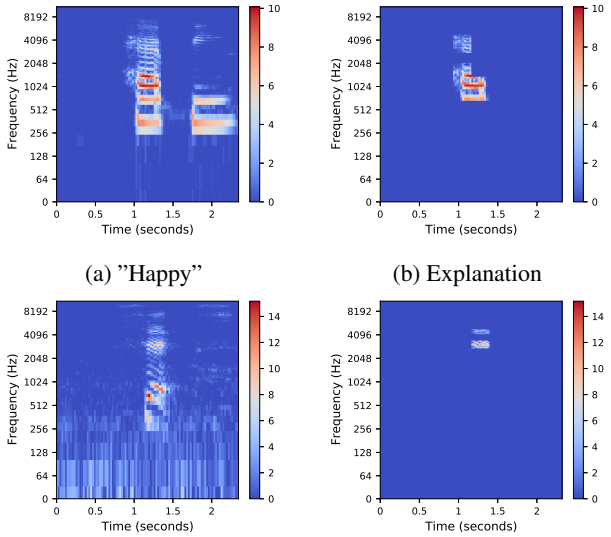


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

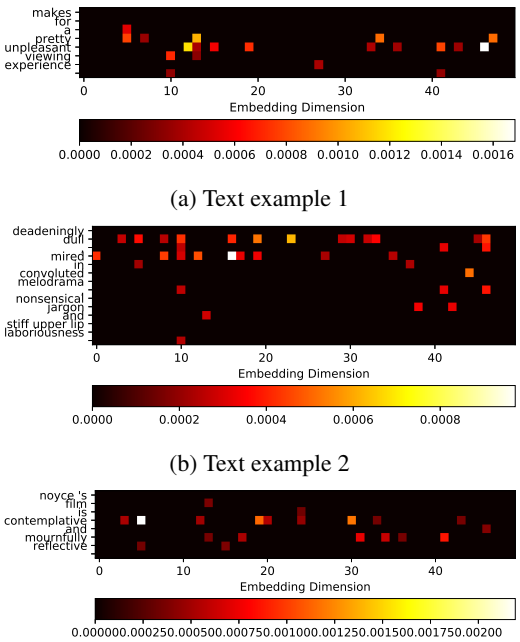
Figure 7: Comparisons between our adversarial explainability approach (Left Column), LIME (Middle Column), and SHAP (Right Column).



(a) "Happy" (b) Explanation

(c) "six" (d) Explanation

Figure 8: Examples for the LeNet speech recognition model.



(a) Text example 1

(b) Text example 2

(c) Text example 3

Figure 9: Examples for the sentence classification model.

5. Additional Examples

We provide additional explanation results of Alexnet image classification model, VGG16 image classification model, ResNet34 image classification model, LeNet speech recognition model, and the sentence classification model in the Appendix G. Due to space limit, in the main paper we only show additional results for explaining the ResNet34 image classification model, Fig. 10.

6. Final Remarks and Conclusion

In this paper, we proposed a new approach for explaining the predictions of DNNs. Interpretability is directly related to the readability of an explanation [11]. An explanation relying on thousands of features is not interpretable. AXAI, similar to LIME, uses input segmentation to create human-readable explanations focused on important input features. Further, AXAI has the following properties,

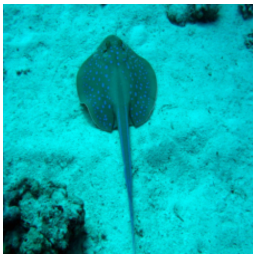
Property 1 (Robustness): Our approach is more robust to the changes in segmentation hyper-parameters in comparison to other segmentation based approaches such as LIME. This is because AXAI does not require a surrogate model trained on "randomly perturbed inputs." AXAI uses the deterministic attack magnitudes as "base explanations" for a given DNN and dataset, and uses segments as an "aid" to visualize the results. The segmentation affects the visualizations. We further explain this in Appendix A. Robustness is identical to stability of explanations as defined in [28]. A

(a) An image of an ostrich

(b) Explanation

(c) An image of a cock

(d) Explanation



(e) An image of a stingray

(f) Explanation

(g) An image of a tench

(h) Explanation

Figure 10: Additional explanation results for a ResNet34 image classification model trained on ImageNet.

lower number of non-deterministic steps in the algorithm enhances stability. A carefully filtered explanation based on our approach simply removes the features that have a low impact on predictions. One can interpret this process as a de-noising step to create a sparse representation of explanations.

Property 2 (Local attribution): Our algorithm is locally stable and uses local attributes to produce explanations. This is because an adversarial attack uses the most minimal amount of noise within an ℓ_2 ball of some small ϵ to fool the DNN. Given the un-targeted nature of the attack used in AXAI, the distributions can be interpreted as estimations of the boundaries among neighboring classes. Thus, one can conclude that the attack magnitudes are a representation of feature contributions to the predictions on a local scale. A similar conclusion is made in [3], where it is argued that gradients can in fact point to important local attributions of a DNN. We explore this in details in Appendix D.

Property 3 (Completeness): Completeness as a property is described as the ability to accurately explain the operations of a DNN [11]. An explanation is more complete when it can explain the behavior of the DNN for a larger set of inputs. [34] and [32] mention the problem of sensitivity and lack of stability in gradient-based algorithms. In the literature, if a solution can reduce the gradient "sensitivity" problem, it can be described as having the "completeness" property [11]. AXAI with PDG attack is complete in the same sense as SmoothGrad is [32]. SmoothGrad takes the average of saliency maps with added Gaussian noise to reduce sensitivity. The PGD attack behaves in a similar manner by adding adversarial noise at each iteration. Both solutions add perturbations to the input to smooth gradient fluctuations. While further research can be done on the power of iterative attacks in their gradient smoothing effects, we argue that AXAI with iterative PGD does have the desirable characteristic and produces stable sharpen visualizations of sensitivity maps for robust explanations.

References

[1] David Alvarez-Melis and Tommi S Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017. 1

[2] David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018. 1, 2

[3] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017. 8

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015. 2

[5] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *arXiv preprint arXiv:1807.03418*, 2018. 2

[6] Michael George Bulmer. *Principles of Statistics*. Courier Corporation, 1979. 4, 13, 15

[7] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *IEEE Symposium on Security and Privacy (SP)*, pages 598–617. IEEE, 2016. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 4, 17

[9] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009. 2

[10] Gil Fidel, Ron Bitton, and Asaf Shabtai. When Explainability Meets Adversarial Learning: Detecting Adversarial Examples using SHAP Signatures. *arXiv preprint arXiv:1909.03418*, 2019. 1

[11] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning, 2018. 7, 8

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015. 3

[13] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019. 2

[14] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. Understanding convolutional neural networks for text classification. *the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018. 2

[15] P Kaur. Convolutional neural networks (CNN) for cifar-10 dataset." parneetk. github. io/blog/cnn-cifar10/, 2017, 2018. 4, 17

[16] Yoon Kim. Convolutional Neural Networks for Sentence Classification, 2014. 6, 15

[17] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6, 2009. 4

[18] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015. 2

[19] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and Understanding Neural Models in NLP, 2015. 6

[20] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017. 1, 2

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2, 3

[22] John H McDonald. *Handbook of Biological Statistics*, volume 2. Sparky House Publishing Baltimore, MD, 2009. 4, 15

[23] Saumitra Mishra, Bob L. Sturm, and Simon Dixon. Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, 2017. 5

[24] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. Adversarial Training Methods for Semi-Supervised Text Classification, 2016. 6

[25] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. A survey of evaluation methods and measures for interpretable machine learning. *arXiv preprint arXiv:1811.11839*, 2018. 1

[26] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005. 6

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016. 1, 2

[28] Marko Robnik-Šikonja and Marko Bohanec. *Perturbation-Based Explanations of Prediction Models*, pages 159–175. 2018. 7

[29] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-Cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 1, 2

[30] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning*, pages 3145–3153. JMLR. Org., 2017. 2

[31] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1

[32] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: Removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2, 8

[33] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, 2014. 1

[34] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the International Conference on Machine Learning*, pages 3319–3328. JMLR. Org., 2017. 2, 8

[35] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *International Conference on Learning Representations (ICLR)*, 2019. 2, 17

[36] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow. SpaceNet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018. 17

[37] Andrea Vedaldi and Stefano Soatto. Quickshift and kernel methods for mode seeking. In *European Conference on Computer Vision*, pages 705–718. Springer, 2008. 3, 11

[38] Pete Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition, 2018. 5, 18

[39] Guannan Zhao, Bo Zhou, Kaiwen Wang, Rui Jiang, and Min Xu. Respond-Cam: Analyzing deep models for 3D imaging data by visualizations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 485–492. Springer, 2018. 2