Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives*

Amit Dhurandhar[†1], Pin-Yu Chen[†1], Ronny Luss[1], Chun-Chen Tu[2], Paishun Ting[2], Karthikeyan Shanmugam[1] and Payel Das[1]

February 22, 2018

Abstract

In this paper we propose a novel method that provides contrastive explanations justifying the classification of an input by a black box classifier such as a deep neural network. Given an input we find what should be minimally and sufficiently present (viz. important object pixels in an image) to justify its classification and analogously what should be minimally and necessarily *absent* (viz. background pixels). We argue that such explanations are natural for humans and are used commonly in domains such as health care and criminology. What is minimally but critically *absent* is an important part of an explanation, which to the best of our knowledge, has not been touched upon by current explanation methods that attempt to explain predictions of neural networks. We validate our approach on three real datasets obtained from diverse domains; namely, a handwritten digits dataset MNIST, a large procurement fraud dataset and an fMRI brain imaging dataset. In all three cases, we witness the power of our approach in generating precise explanations that are also easy for human experts to understand and evaluate.

1 Introduction

Steve is the tall guy with long hair who does not wear glasses. Explanations as such are used frequently by people to identify other people or items of interest. We see in this case that characteristics such as being tall and having long hair help describe the person, although incompletely. The absence of glasses is important to complete the identification and help distinguish him from, for instance, Bob who is tall, has long hair and wears glasses. It is common for us humans to state such contrastive facts when we want to accurately explain something.

*† implies equal contribution. 1 and 2 indicate affiliations to IBM Research and University of Michigan respectively.

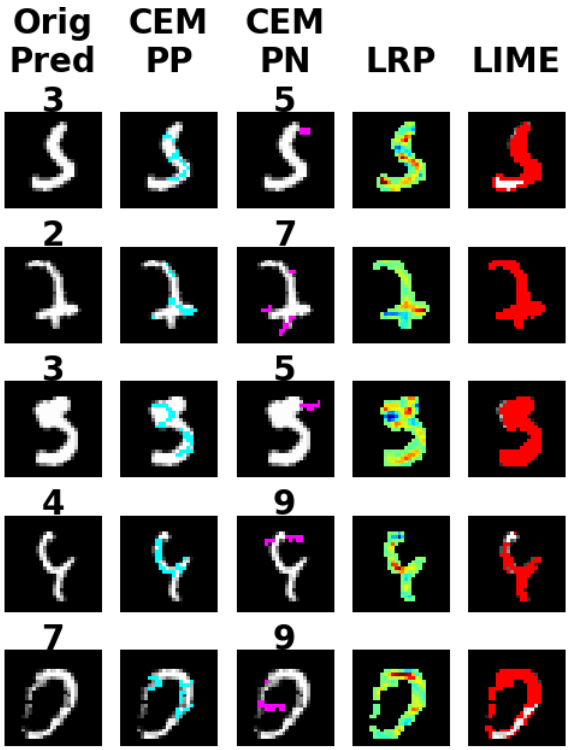

Figure 1: Comparison of our CEM versus LRP and LIME on MNIST. PP/PN stands for Pertinent Positive/Negative. The color scheme is as follows: For CEM pertinent positives, the minimal that should be present is cyan. For PN, the minimal that should be absent is pink and corresponding digit labels are the number it would become if the pink were present. For LRP, green is neutral, red/yellow is positive relevance, and blue is negative relevance. For LIME, red is positive relevance and white is neutral.

These contrastive facts are by no means a list of all possible characteristics that should be absent in an input to distinguish it from all other classes that it does not belong to, but rather a minimal set of characteristics/features that help distinguish it from the "closest" class that it does not belong to.

In this paper we want to generate such explanations for deep networks, in which, besides highlighting what is minimally sufficient (e.g. tall and long hair) in an input to justify its classification, we also want to identify contrastive characteristics or features that should be minimally and critically *absent* (e.g. glasses), so as to maintain the current classification and to distinguish it from another input that is "closest" to it but would be classified differently (e.g. Bob). We thus want to generate explanations of the form, "*An input x is classified in class y because features $f_i, \cdots, f_k$ are present and because features $f_m, \cdots, f_p$ are absent.*" The need for such an aspect as what constitutes a good explanation,

which is essentially a counterfactual, has been stressed on recently [11]. It may seem that such crisp explanations are only possible for binary data. However, they are also applicable to continuous data. The presence (or absence) of features corresponds to a signal being present for those features, which in the case of images would be features that are non-zero for an input or those that represent objects, as opposed to say background pixels in an image. What corresponds to a signal therefore is data specific. For example, in Figure 1, where we see hand-written digits from MNIST [37] dataset, the black background represents no signal or absence of those specific features, which in this case are pixels with a value of zero. Any non-zero value then would indicate the presence of those features/pixels. Although one may argue that there is some information loss in our form of explanation, we believe that such explanations are lucid and easily understandable by humans who can always further delve into the details of our generated explanations such as the precise feature values, which are readily available.

In fact, there is a strong motivation to have such form of explanations due to their presence in certain human-critical domains. In medicine and criminology there is the notion of pertinent positives and pertinent negatives [15], which together constitute a complete explanation. A pertinent positive is a factor whose *presence* is minimally sufficient in justifying the final classification. On the other hand, a pertinent negative is a factor whose *absence* is necessary in asserting the final classification. For example in medicine, a patient showing symptoms of cough, cold and fever, but no sputum or chills, will most likely be diagnosed as having flu rather than having pneumonia. Cough, cold and fever are pertinent positives for both flu and pneumonia, however, the absence of sputum and chills leads to the diagnosis of flu. Thus, sputum and chills are pertinent negatives, which along with the pertinent positives are critical and in some sense sufficient for an accurate diagnosis. An example in criminology would be a murder scene where, at the victims residence, we find that his safe is empty with all his valuables missing. The empty safe is a pertinent negative and could indicate theft being the primary motive as opposed to murder by an acquaintance. Hence, pertinent negatives can be viewed as factors whose absence is critical in determining the correct class for an example and distinguishing it from the "closest" other class.

We thus propose an explanation method called *contrastive explanations method* (CEM) for deep neural networks that highlights not only the pertinent positives but also the pertinent negatives. This is seen in Figure 1 where our explanation of the image being predicted as a 3 in the first row does not only highlight the important pixels (which look like a 3) that should be present for it to be classified as a 3, but also highlights a small horizontal line (the pertinent negative) at the top whose presence would change the classification of the image to a 5 and thus should be absent for the classification to remain a 3. Therefore, our explanation for the digit in row 1 of Figure 1 to be a 3 would be: *The row 1 digit is a 3 because the cyan pixels (shown in column 2) are present and the pink pixels (shown in column 3) are absent.* This second part is critical for an accurate classification and is *not* highlighted by any of the other state-of-the-art

interpretability methods such as layerwise relevance propagation (LRP) [1] or locally interpretable model-agnostic explanations (LIME) [29], for which the respective results are shown in columns 4 and 5 of Figure 1. Moreover, given the original image, our pertinent positives highlight what should be present that is necessary and sufficient for the example to be classified as a 3. This is not the case for the other methods, which essentially highlight positively or negatively relevant pixels that may not be necessary or sufficient to justify the classification.

Pertinent Negatives vs Negatively Relevant Features: Another important thing to note here is the conceptual distinction between pertinent negatives that we identify and negatively correlated or relevant features that other methods highlight. The question we are trying to answer is: *why is input x classified in class y?*. Ergo, any human asking this question wants all the evidence in support of the hypothesis of x being classified as class y. Our pertinent positives as well as negatives are evidences in support of this hypothesis. However, unlike the positively relevant features highlighted by other methods that are also evidence supporting this hypothesis, the negatively relevant features by definition do not. Hence, another motivation for our work is that we believe when a human asks the above question, they are more interested in evidence supporting the hypothesis rather than information that devalues it. This latter information is definitely interesting, but is of secondary importance when it comes to understanding the human's intent behind the question.

Given an input and its classification by a neural network, CEM creates explanations for it as follows:

1. It finds a minimal amount of features in the input that are sufficient in themselves to yield the same classification. We do this by solving a (novel) constrained optimization problem that perturbs the input such that when a minimal number of non-zero features (viz. non-background/object pixels) are made or brought closer to zero (or background), i.e. are *deleted*, these deleted features (on their own drawn as a distinct image) maintain the original classification. In other words, they are pertinent positives that are *sufficient* evidence for the original classification of the input.

2. It also finds a minimal amount of features that should be *absent* in the input to prevent the classification result from changing. We do this by solving another (novel) constrained optimization problem that perturbs the input such that a minimal number of zero valued features (viz. background pixels) when made non-zero, i.e. when *added*, cause the neural network classification to change. These added features are the pertinent negatives such that their absence is *necessary* for the prediction of the input.

3. It does (1) and (2) "close" to the data manifold so as to obtain more "realistic" explanations. We do this by using a convolutional autoencoder (CAE) as a regularizer for our objectives in (1) and (2). The data manifold is learned using a CAE which has been recently shown to be state-of-the-art [24] compared with other types of autoencoders.

We enhance our methods to do (3), so that the resulting explanations are more likely to be close to the true data manifold and thus match human intuition rather than arbitrary perturbations that may change the classification. Of course, learning a good representation using an autoencoder may not be possible in all situations due to limitations such as insufficient data or bad data quality. It also may not be necessary in some cases if all combinations of feature values have semantics in the domain.

The amount of perturbation depends on the metric used. We chose to use an elastic net regularizer to measure the amount of distortion, since it has been shown recently [7] to produce the best or sharpest adversarial examples over other norms. This fits well with our form of explanations which focus on the presence or the absence of features. Our method proposed in Section 3 thus has close relations to methods proposed for adversarial attacks on neural networks [5, 7] with certain key differences. The main one being that the (untargeted) attack methods are largely unconstrained where additions and deletions are performed simultaneously to alter the classification of the input. Our method, on the other hand, obtains pertinent positives and pertinent negatives that are sufficient and necessary, respectively, to justify the classification by enforcing the applied perturbations to be deletions only or additions only. Moreover, our optimization objective is itself distinct as we are searching for features that are minimally sufficient in themselves to maintain the original classification. As such, our work demonstrates how attack methods can be adapted to create explanation methods, and thus exhibits the strong connection between the two. Finally, we note that the integration of an autoencoder into a perturbation generation method for more "realistic" and more intuitive explanations has been largely unexplored previously.

We validate our approaches on three real-world datasets. The first one is the hand-written digits dataset MNIST [37], from which we generate explanations with and without an autoencoder. The second is a procurement fraud dataset from a large corporation containing millions of invoices from tens of thousands of vendors that have different risk levels. The third one is a brain functional MRI (fMRI) imaging dataset from the publicly accessible Autism Brain Imaging Data Exchange (ABIDE) I database [10], which comprises of resting-state fMRI acquisitions of subjects diagnosed with autism spectrum disorder (ASD) and neurotypical individuals. For the latter two cases, we do not consider using autoencoders. This is because the fMRI dataset is insufficiently large especially given its high-dimensionality. For the procurement data on the other hand, all combination of allowed feature values are (intuitively) reasonable. The preceding characteristics make autoencoders less suitable for these datasets. In all three cases, we witness the power of our approach in creating more precise explanations that also match human judgment.

2 Related Work

Researchers have put great efforts in devising frameworks and algorithms for interpretable modeling. Examples include establishment for rule/decision lists [33, 36], prototype exploration [13,19], developing methods inspired by psychometrics [17] and learning human-consumable models [6]. There are also works [29] which focus on answering instance-specific user queries by locally approximating a high-performing complex model using a simpler but easy-to-understand one, which could be used to gain confidence in the complex model. There is also a recent work [30] that proposes a unified approach to create local model explanations with certain desirable properties that many current methods seem to lack. Moreover, there is also some interesting work which tries to formalize and quantify interpretability [9].

A recent survey [23] looks primarily at two methods for understanding neural networks: a) Methods [25, 27] that produce a prototype for a given class by optimizing the confidence score for the class subject to some regularization on the prototype, b) Explaining a neural network's decision on an image by highlighting relevant parts using a technique called Layer-wise relevance propagation [1]. This technique starts from the last layer and progressively assigns weights to neurons of layers below connected to a single neuron on a layer above satisfying some weight conservation properties across layers. We observe that type (b) methods are local model explanations on a specific image while type (a) methods are more global producing prototypes for a given class. Other works also investigate methods of the type (b) discussed above for vision [31, 32] and NLP applications [21].

Most of the these explanation methods, however, focus on features that are present, even if they may highlight negatively contributing features to the final classification. As such, they do not identify features that should be necessarily and sufficiently present or absent to justify for an individual example its classification by the model.

3 Contrastive Explanations Method

This section details the proposed contrastive explanations method. Let $\mathcal{X}$ denote the feasible data space and let $(\mathbf{x}_0, t_0)$ denote an example $\mathbf{x}_0 \in \mathcal{X}$ and its inferred class label t_0 obtained from a neural network model. The modified example $\mathbf{x} \in \mathcal{X}$ based on $\mathbf{x}_0$ is defined as $\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is a perturbation applied to $\mathbf{x}_0$. Our method of finding pertinent positives/negatives is formulated as an optimization problem over the perturbation variabale $\boldsymbol{\delta}$ that is used to explain the model's prediction results. We denote the prediction of the model on the example $\mathbf{x}$ by $\text{Pred}(\mathbf{x})$, where $\text{Pred}(\cdot)$ is any function that outputs a vector of prediction scores for all classes, such as prediction probabilities, logits (unnormalized probability) widely used in neural networks, among others.

To ensure the modified example $\mathbf{x}$ is still close to the data manifold of natural examples, we propose to use an autoencoder to evaluate the closeness of $\mathbf{x}$ to the data manifold. An autoencoder consists of an encoder that compresses

high-dimensional input data to low-dimensional representations, and a decoder that reconstructs the input data based on their low-dimensional representations. We denote by $\mathrm{AE}(\mathbf{x})$ the reconstructed example of $\mathbf{x}$ using the autoencoder $\mathrm{AE}(\cdot)$.

3.1 Finding Pertinent Negatives

For pertinent negative analysis, one is interested in what is missing in the model prediction. For any natural example $\mathbf{x}_0$, we use the notation $\mathcal{X}/\mathbf{x}_0$ to denote the space of missing parts with respect to $\mathbf{x}_0$. We aim to find an interpretable perturbation $\boldsymbol{\delta} \in \mathcal{X}/\mathbf{x}_0$ to study the difference between the most probable class predictions in $\arg\max_i[\mathrm{Pred}(\mathbf{x}_0)]_i$ and $\arg\max_i[\mathrm{Pred}(\mathbf{x}_0+\boldsymbol{\delta})]_i$. Given $(\mathbf{x}_0, t_0)$, our method finds a pertinent negative by solving the following optimization problem:

$$\min_{\boldsymbol{\delta}\in\mathcal{X}/\mathbf{x}_0} \; c\cdot f_\kappa^{\mathrm{neg}}(\mathbf{x}_0,\boldsymbol{\delta}) + \beta\|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\delta}\|_2^2 + \gamma\|\mathbf{x}_0+\boldsymbol{\delta}-\mathrm{AE}(\mathbf{x}_0+\boldsymbol{\delta})\|_2^2. \quad (1)$$

We elaborate on the role of each term in the objective function (1) as follows. The first term $f_\kappa^{\mathrm{neg}}(\mathbf{x}_0,\boldsymbol{\delta})$ is a designed loss function that encourages the modified example $\mathbf{x} = \mathbf{x}_0+\boldsymbol{\delta}$ to be predicted as a different class than $t_0 = \arg\max_i[\mathrm{Pred}(\mathbf{x}_0)]_i$. The loss function is defined as:

$$f_\kappa^{\mathrm{neg}}(\mathbf{x}_0,\boldsymbol{\delta}) = \max\{[\mathrm{Pred}(\mathbf{x}_0+\boldsymbol{\delta})]_{t_0} - \max_{i\neq t_0}[\mathrm{Pred}(\mathbf{x}_0+\boldsymbol{\delta})]_i, -\kappa\} \quad (2)$$

where $[\mathrm{Pred}(\mathbf{x}_0+\boldsymbol{\delta})]_i$ is the i-th class prediction score of $\mathbf{x}_0+\boldsymbol{\delta}$. The hinge-like loss function favors the modified example $\mathbf{x}$ to have a top-1 prediction class different from that of the original example $\mathbf{x}_0$. The parameter $\kappa \geq 0$ is a confidence parameter that controls the separation between $[\mathrm{Pred}(\mathbf{x}_0+\boldsymbol{\delta})]_{t_0}$ and $\max_{i\neq t_0}[\mathrm{Pred}(\mathbf{x}_0+\boldsymbol{\delta})]_i$. The second and the third terms $\beta\|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\delta}\|_2^2$ in (1) are jointly called the elastic net regularizer, which is used for efficient feature selection in high-dimensional learning problems [40]. The last term $\|\mathbf{x}_0+\boldsymbol{\delta}-\mathrm{AE}(\mathbf{x}_0+\boldsymbol{\delta})\|_2^2$ is an L_2 reconstruction error of $\mathbf{x}$ evaluated by the autoencoder. This is relevant provided that a well-trained autoencoder for the domain is obtainable. The parameters $c, \beta, \gamma, \geq 0$ are the associated regularization coefficients.

3.2 Finding Pertinent Positives

For pertinent positive analysis, we are interested in the critical features that are readily present in the input. Given a natural example $\mathbf{x}_0$, we denote the space of its existing components by $\mathcal{X}\cap\mathbf{x}_0$. Here we aim at finding an interpretable perturbation $\boldsymbol{\delta} \in \mathcal{X}\cap\mathbf{x}_0$ such that after removing it from $\mathbf{x}_0$, $\arg\max_i[\mathrm{Pred}(\mathbf{x}_0)]_i = \arg\max_i[\mathrm{Pred}(\boldsymbol{\delta})]_i$. That is, $\mathbf{x}_0$ and $\boldsymbol{\delta}$ will have the same top-1 prediction class t_0, indicating that the removed perturbation $\boldsymbol{\delta}$ is representative of the model prediction on $\mathbf{x}_0$. Similar to finding pertinent negatives, we formulate finding pertinent positives as the following optimization problem:

$$\min_{\boldsymbol{\delta}\in\mathcal{X}\cap\mathbf{x}_0} \; c\cdot f_\kappa^{\mathrm{pos}}(\mathbf{x}_0,\boldsymbol{\delta}) + \beta\|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\delta}\|_2^2 + \gamma\|\boldsymbol{\delta}-\mathrm{AE}(\boldsymbol{\delta})\|_2^2, \quad (3)$$

Algorithm 1 Contrastive Explanations Method (CEM)

Input: example (x_0, t_0), neural network model $\mathcal{N}$ and (optionally) an autoencoder AE

1) Solve (1) and obtain,
$\boldsymbol{\delta}^{\text{neg}} \leftarrow \text{argmin}_{\boldsymbol{\delta} \in \mathcal{X}/\mathbf{x}_0} \; c \cdot f_\kappa^{\text{neg}}(\mathbf{x}_0, \boldsymbol{\delta}) + \beta\|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\delta}\|_2^2 + \gamma\|\mathbf{x}_0 + \boldsymbol{\delta} - \text{AE}(\mathbf{x}_0 + \boldsymbol{\delta})\|_2^2$.

2) Solve (3) and obtain,
$\boldsymbol{\delta}^{\text{pos}} \leftarrow \text{argmin}_{\boldsymbol{\delta} \in \mathcal{X} \cap \mathbf{x}_0} \; c \cdot f_\kappa^{\text{pos}}(\mathbf{x}_0, \boldsymbol{\delta}) + \beta\|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\delta}\|_2^2 + \gamma\|\boldsymbol{\delta} - \text{AE}(\boldsymbol{\delta})\|_2^2$.

{γ in both Step 1 and Step 2 can be set to 0 if an AE is not used.} {Strategies to solve (1) and (3) are described in Section 3.3.}

return $\boldsymbol{\delta}^{\text{pos}}$ and $\boldsymbol{\delta}^{\text{neg}}$. {Our Explanation: Input x_0 is classified as class t_0 because features $\boldsymbol{\delta}^{\text{pos}}$ are present and because features $\boldsymbol{\delta}^{\text{neg}}$ are absent. Code is available at `https://github.com/chunchentu/CEM/blob/master/README.md`}

where the loss function $f_\kappa^{\text{pos}}(\mathbf{x}_0, \boldsymbol{\delta})$ is defined as

$$f_\kappa^{\text{pos}}(\mathbf{x}_0, \boldsymbol{\delta}) = \max\{\max_{i \neq t_0}[\text{Pred}(\boldsymbol{\delta})]_i - [\text{Pred}(\boldsymbol{\delta})]_{t_0}, -\kappa\}. \tag{4}$$

In other words, for any given confidence $\kappa \geq 0$, the loss function f_κ^{pos} is minimized when $[\text{Pred}(\boldsymbol{\delta})]_{t_0}$ is greater than $\max_{i \neq t_0}[\text{Pred}(\boldsymbol{\delta})]_i$ by at least κ.

3.3 Algorithmic Details

We apply a projected fast iterative shrinkage-thresholding algorithm (FISTA) [2] to solve problems (1) and (3). FISTA is an efficient solver for optimization problems involving L_1 regularization. Take pertinent negative as an example, assume $\mathcal{X} = [-1, 1]^p$, $\mathcal{X}/\mathbf{x}_0 = [0, 1]^p/\mathbf{x}_0$ and let $g(\boldsymbol{\delta}) = f_\kappa^{\text{neg}}(\mathbf{x}_0, \boldsymbol{\delta}) + \|\boldsymbol{\delta}\|_2^2 + \gamma\|\mathbf{x}_0 + \boldsymbol{\delta} - \text{AE}(\mathbf{x}_0 + \boldsymbol{\delta})\|_2^2$ denote the objective function of (1) without the L_1 regularization term. Given the initial iterate $\boldsymbol{\delta}^{(0)} = \mathbf{0}$, projected FISTA iteratively updates the perturbation I times by

$$\boldsymbol{\delta}^{(k+1)} = \Pi_{[0,1]^p}\{S_\beta(\mathbf{y}^{(k)} - \alpha_k \nabla g(\mathbf{y}^{(k)}))\}; \tag{5}$$

$$\mathbf{y}^{(k+1)} = \Pi_{[0,1]^p}\{\mathbf{x}^{(k+1)} + \frac{k}{k+3}(\boldsymbol{\delta}^{(k+1)} - \boldsymbol{\delta}^{(k)})\}, \tag{6}$$

where $\Pi_{[0,1]^p}$ denotes the vector projection onto the set $\mathcal{X}/\mathbf{x}_0 = [0, 1]^p$, α_k is the step size, $\mathbf{y}^{(k)}$ is a slack variable accounting for momentum acceleration with $\mathbf{y}^{(0)} = \boldsymbol{\delta}^{(0)}$, and $S_\beta : \mathbb{R}^p \mapsto \mathbb{R}^p$ is an element-wise shrinkage-thresholding

function defined as

$$[S_\beta(\mathbf{z})]_i = \begin{cases} \mathbf{z}_i - \beta, & \text{if } \mathbf{z}_i > \beta; \\ 0, & \text{if } |\mathbf{z}_i| \leq \beta; \\ \mathbf{z}_i + \beta, & \text{if } \mathbf{z}_i < -\beta, \end{cases} \tag{7}$$

for any $i \in \{1, \ldots, p\}$. The final perturbation $\boldsymbol{\delta}^{(k^*)}$ for pertinent negative analysis is selected from the set $\{\boldsymbol{\delta}^{(k)}\}_{k=1}^{I}$ such that $f_\kappa^{\text{pos}}(\mathbf{x}_0, \boldsymbol{\delta}^{(k^*)}) = 0$ and $k^* = \arg\min_{k \in \{1,\ldots,I\}} \beta\|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\delta}\|_2^2$. A similar projected FISTA optimization approach is applied to pertinent positive analysis.

Eventually, as seen in Algorithm 1, we use both the pertinent negative $\boldsymbol{\delta}^{\text{neg}}$ and the pertinent positive $\boldsymbol{\delta}^{\text{pos}}$ obtained from our optimization methods to explain the model prediction. The last term in both (1) and (3) will be included only when an accurate autoencoder is available, or otherwise γ is to be set to zero.

4 Experiments

This section provides experimental results on three representative datasets, including the handwritten digits dataset MNIST, a procurement fraud dataset obtained from a large corporation having millions of invoices and tens of thousands of vendors, and a brain imaging fMRI dataset containing brain activity patterns for both normal and autistic individuals. We compare our approach with previous state-of-the-art methods and demonstrate our superiority in being able to generate more accurate and intuitive explanations.

As to the implementation of the projected FISTA for finding pertinent negatives and pertinent positives, we set the regularization coefficients $\beta = 0.1$, and $\gamma = \{0, 100\}$. The parameter c is set to 0.1 initially, and is searched for 9 times guided by run-time information. In each search, if f_κ never reaches 0, then in the next search, c is multiplied by 10, otherwise it is averaged with the current value for the next search. For each search in c, we run $I = 1000$ iterations using the SGD solver provided by TensorFlow. The initial learning rate is set to be 0.01 with a square-root decaying step size. The best perturbation among all searches is used as the pertinent positive/negative for the respective optimization problems.

4.1 Handwritten Digits

We first report results on the handwritten digits MNIST dataset. In this case, we provide examples of explanations for our method with and without an autoencoder.

4.1.1 Setup

The handwritten digits are classified using a feed-forward convolutional neural network (CNN) trained on 60,000 training images from the MNIST benchmark

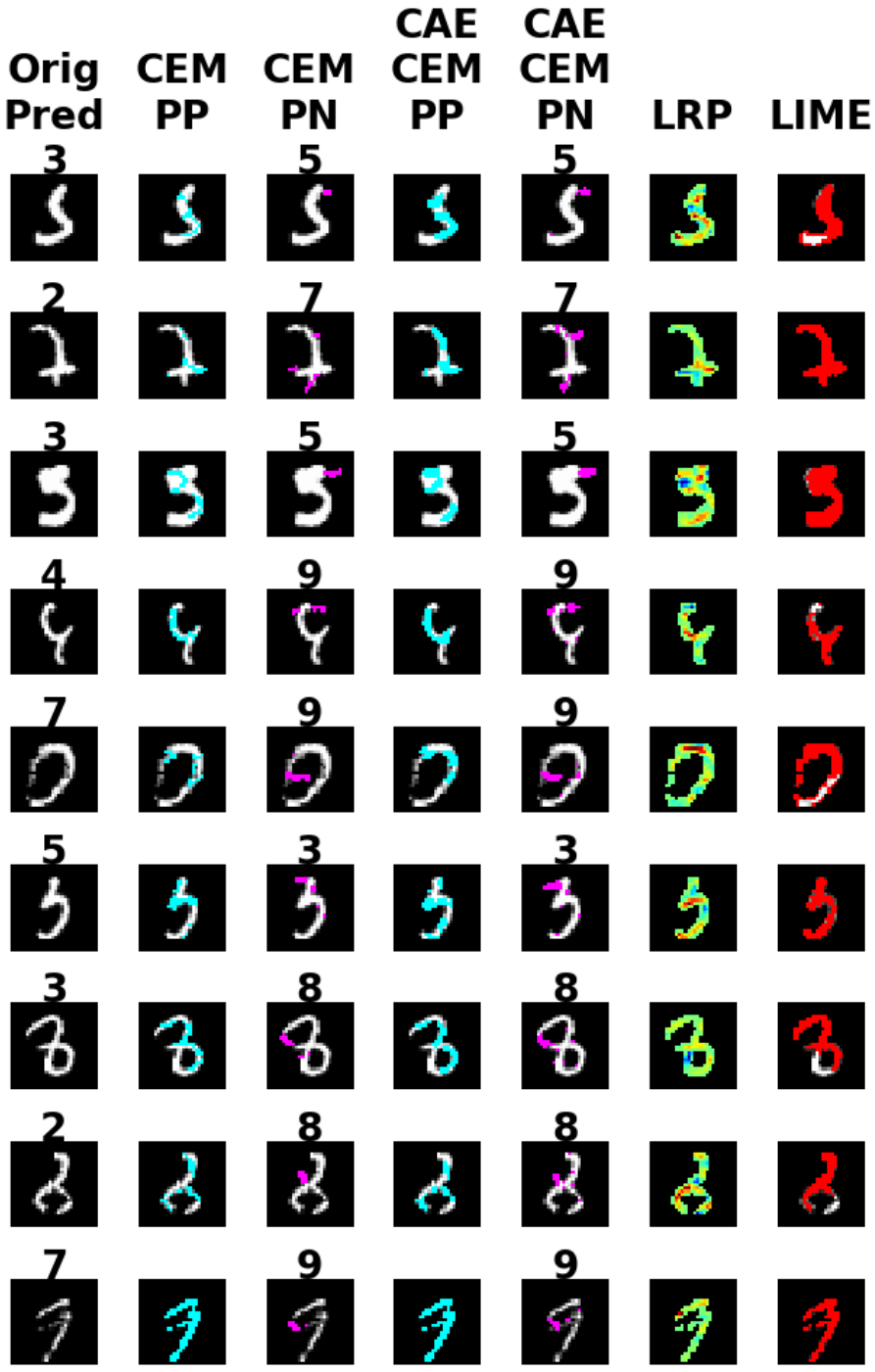

Figure 2: Comparison of our Contrastive Explanations Methods versus LRP and LIME on MNIST. CAE means a convolutional autoencoder was used. The color scheme is as follows: For CEM pertinent positives, the minimal that should be present is cyan. For pertinent negatives, the minimal that should be absent is pink and corresponding digit labels are the number it would become if the pink were present. For LRP, green is neutral, red/yellow is positive relevance, and blue is negative relevance. For LIME, red is positive relevance and white is neutral.

dataset. The CNN has two sets of convolution-convolution-pooling layers, followed by three fully-connected layers. All the convolution layers use a ReLU activation function, while the pooling layers use a 2×2 max-pooling kernel to downsample each feature map from their previous layer. In the first set, both the convolution layers contain 32 filters, each using a $3 \times 3 \times D$ kernel, where D is an appropriate kernel depth. Both the convolution layers in the second set, on the other hand, contain 64 filters, each again using a $3 \times 3 \times D$ kernel. The three fully-connected layers have 200, 200 and 10 neurons, respectively. The test accuracy of the CNN is around 99.4%. The logits of this CNN are used as model prediction scores, and we set $\mathcal{X} = [-0.5, 0.5]^p$, $\mathcal{X}/\mathbf{x}_0 = [0, 0.5]^p/\mathbf{x}_0$ and $\mathcal{X} \cap \mathbf{x}_0 = [0, 0.5]^p \cap \mathbf{x}_0$ for any natural example $\mathbf{x}_0 \in \mathcal{X}$.

The CAE architecture contains two major components: an encoder and a decoder. The encoder compresses the 28×28 input image down to a 14×14 feature map using the architecture of convolution-convolution-pooling-convolution. Both of the first two convolution layers contain 16 filters, each using a $3 \times 3 \times D$ kernel, where D is again an appropriate kernel depth. They also incorporate a ReLU activation function in them. The pooling layer is of the max-pooling type with a 2×2 kernel. The last convolution layer has no activation function, but instead has a single filter with a $3 \times 3 \times D$ kernel. The decoder, on the other hand, recovers an image of the original size from the feature map in the latent space. It has an architecture of convolution-upsampling-convolution-convolution. Again, both of the first two convolution layers have a ReLU activation function applied to the outputs of the 16 filters, each with a $3 \times 3 \times D$ kernel. The upsampling layer enlarges its input feature maps by doubling their side length through repeating each pixel four times. The last convolution layers has a single filter with the kernel size $3 \times 3 \times D$.

4.1.2 Results

Our CEM method is applied to MNIST with a variety of examples illustrated in Figure 2. In addition to what was shown in Figure 1 in the introduction, results using a convolutional autoencoder (CAE) to learn the pertinent positives and negatives are displayed. While results without an CAE are quite convincing, the CAE clearly improves the pertinent positives and negatives in many cases. Regarding pertinent positives, the cyan highlighted pixels in the column with CAE (CAE CEM PP) are a superset to the cyan-highlighted pixels in column without (CEM PP). While these explanations are at the same level of confidence regarding the classifier, explanations using an AE are visually more interpretable. Take for instance the digit classified as a 2 in row 2. A small part of the tail of a 2 is used to explain the classifier without a CAE, while the explanation using a CAE has a much thicker tail and larger part of the vertical curve. In row 3, the explanation of the 3 is quite clear, but the CAE highlights the same explanation but much thicker with more pixels. The same pattern holds for pertinent negatives. The horizontal line in row 4 that makes a 4 into a 9 is much more pronounced when using a CAE. The change of a predicted 7 into a 9 in row 5 using a CAE is much more pronounced. The other rows exhibit similar

patterns.

The two state-of-the-art methods we use for explaining the classifier in Figure 2 are LRP and LIME. LRP experiments used the toolbox from [20] and LIME code was adapted from `https://github.com/marcotcr/lime`. LRP has a visually appealing explanation at the pixel level. Most pixels are deemed irrelevant (green) to the classification (note the black background of LRP results was actually neutral). Positively relevant pixels (yellow/red) are mostly consistent with our pertinent positives, though the pertinent positives do highlight more pixels for easier visualization. The most obvious such examples are row 3 where the yellow in LRP outlines a similar 3 to the pertinent positive and row 6 where the yellow outlines most of what the pertinent positive provably deems necessary for the given prediction. There is little negative relevance in these examples, though we point out two interesting cases. In row 4, LRP shows that the little curve extending the upper left of the 4 slightly to the right has negative relevance (also shown by CEM as not being positively pertinent). Similarly, in row 3, the blue pixels in LRP are a part of the image that must obviously be deleted to see a clear 3. LIME is also visually appealing. However, the results are based on superpixels - the images were first segmented and relevant segments were discovered. This explains why most of the pixels forming the digits are found relevant. While both methods give important intuitions, neither illustrate what is necessary and sufficient about the classifier results as does our contrastive explanations method.

4.2 Procurement Fraud

In this experiment, we evaluated our methods on a real procurement dataset obtained from a large corporation. This nicely complements our other experiments on image datasets.

4.2.1 Setup

The data spans a one-year period and consists of millions of invoices submitted by over tens of thousands vendors across 150 countries. The invoices were labeled as being either low risk, medium risk, or high risk based on a large team that approves these invoices. To make such an assessment, besides just the invoice data, we and the team had access to multiple public and private data sources such as vendor master file (VMF), risky vendors list (RVL), risky commodity list (RCL), financial index (FI), forbidden parties list (FPL) [4, 34], country perceptions index (CPI) [18], tax havens list (THL) and Dun & Bradstreet numbers (DUNs) [3]. The VMF has information such as names of the vendors registered with the company, their addresses, account numbers and date of registration. The RVL and RCL contain lists of potentially fraudulent vendors and commodities that are often easy to manipulate. The FI contains information such as maturity of a vendor and their stock trends. The FPL released by the US government every year has two lists of suspect businesses. The CPI is a public source scoring (0-100) the risk of doing business in a particular country.

Invoice ID	Risk Level	Events Triggered	Pertinent Positives	Pertinent Negatives	Expert Feedback
Anon-1	Low	1, 2, 9	2, 9	7	... vendor being registered and having a DUNs number makes the invoice low risk. However, if it came from a low CPI country then the risk would be uplifted given that the invoice amount is already high.
Anon-2	Medium	2, 4, 7	2, 4	6	... the vendor being registered with the company keeps the risk manageable given that it is a risky commodity code. Nonetheless, if he was part of any of the FPL lists the invoice would most definitely be blocked.
Anon-3	High	1, 4, 5, 11	1, 4, 11	2, 9	... the high invoice amount, the risky commodity code and no physical address makes this invoice high risk. The risk level would definitely have been somewhat lesser if the vendor was registered in VMF and DUNs.

Table 1: Above we see 3 example invoices (IDs anonymized), one at low risk level, one at medium and one at high risk level. The corresponding pertinent positives and negatives are highlighted by our method. We also report feedback from a human expert in the last column, which validates the quality of our explanations. The numbers that the events correspond to are given in Section 4.2.1.

The lower the CPI for a country, the worse the perception and hence higher the risk. Tax havens are countries such as the Cayman Islands where the taxes are minimal and complete privacy is maintained regarding people's financials. Dun & Bradstreet offers a unique DUNS number and DUNS name for each business registered with them. A DUNS ID provides a certain level of authenticity to the business.

Based on the above data sources, there are tens of features and events whose occurrence hints at the riskiness of an invoice. Here are some representative ones. 1) if the spend with a particular vendor is significantly higher than with other vendors in the same country, 2) if a vendor is registered with a large corporation and thus its name appears in VMF, 3) if a vendor belongs to RVL, 4) if the commodity on the invoice belongs to RCL, 5) if the maturity based on FI is low, 6) if vendor belongs to FPL, 7) if a vendor is in a high risk country (i.e. CPI < 25), 8) if a vendor or its bank account is located in a tax haven, 9) if a vendor has a DUNs number, 10) if a vendor and the employee bank account numbers match, 11) if a vendor only possesses a PO box with no street address.

With these data, we trained a three-layer neural network with fully connected layers, 512 rectified linear units and a three-way softmax function. The ten-fold cross validation accuracy of the network is high (91.6%). The domain experts, however, wanted to know the reasons its decisions are based on, which could help them get a flavor of how the network works and if they could trust it.

In this experiment, we didn't use the CAE regularization since all combinations of feature values are possible and there are no semantic/intuitive inconsistencies that crop up if certain features are turned on or off in the presence of others. This is different from the MNIST case, where a purely random perturbation can add or delete pixels that result in poor semantic interpretation.

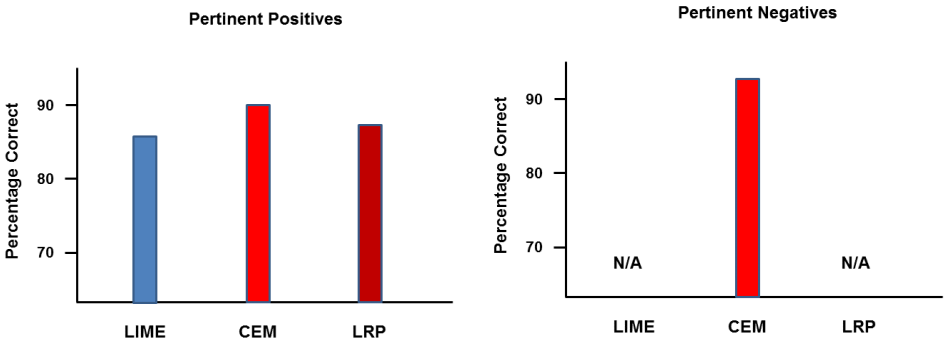

Figure 3: Above we see the fraction of invoices on which the explanations of the different methods were deemed appropriate by experts. Only our method produces pertinent negatives, which are quite effective as is seen in the figure on the right. Our pertinent positives (left figure) are also more accurate than the other methods for whom we picked positively relevant features as a proxy for pertinent positives.

4.2.2 Results

With the help of domain experts, we evaluated the different explanation methods. We randomly chose 15 invoices that were classified as low risk, 15 classified as medium risk and 15 classified as high risk. We asked for feedback on these 45 invoices in terms of whether or not the pertinent positives and pertinent negatives (only ours) highlighted by each of the methods was suitable to produce the classification. To evaluate each method, we computed the percentage of invoices with explanations agreed by the experts based on this feedback.

In Figure 3, we see the percentage of times the pertinent positives (left figure) matched with the experts judgment for the different methods as well as additionally the pertinent negatives (right figure) for our method. We used positive relevance as a proxy for pertinent positives for the competing methods. As we have argued before, the negatively relevant features are not conceptually the same as pertinent negatives and hence, there is no appropriate proxy for them. This is why we show not-applicable (N/A) for those methods. We observe

that in both cases our explanations closely match human judgment and are superior to its competitors.

Table 1 shows 3 example invoices, one belonging to each class and the explanations produced by our method along with the expert feedback. We see that the expert feedback validates our explanations and showcases the power of pertinent negatives in making the explanations more complete as well as intuitive to reason with. An interesting aspect here is that the medium risk invoice could have been perturbed towards low risk or high risk. However, our method found that it is closer (minimum perturbation) to being high risk and thus suggested a pertinent negative that takes it into that class. *Such informed decisions can be made by our method as it searches for the most "crisp" explanation, arguably similar to those of humans.*

4.3 Brain Functional Imaging

In this experiment we look at explaining why a certain individual was classified as autistic as opposed to a normal/typical individual.

4.3.1 Setup

The brain imaging dataset employed in this study is the Autism Brain Imaging Data Exchange (ABIDE) I [10], a large publicly available dataset consisting of resting-state fMRI acquisitions of subjects diagnosed with autism spectrum disorder (ASD), as well as of neuro-typical individuals. Resting state fMRI provides neural measurements of the functional relationship between brain regions and is particularly useful for investigating clinical populations. Previously preprocessed acquisitions were downloaded (http://preprocessedconnectomes-project.org/abide/). We used the C-PAC preprocessing pipeline which included slice-time correction, motion correction, skull-stripping, and nuisance signal regression. Functional data was band-pass filtered (0.01—0.1 Hz) and spatially registered using a nonlinear method to a template space (MNI152). We limited ourselves to acquisitions with repetition time of 2s (sites NYU, SDSU, UM, USM) that were included in the original study of Di Martino et al. [10] and that passed additional manual quality control, resulting in a total of 147 ASD and 146 typical subjects (right-handed male, average age 16.5 yr). The CC200 functional parcellation atlas [8] of the brain, totaling 200 regions, was used to estimate the brain connectivity matrix. The mean time series for regions of interest (ROI) was extracted for each subject. A Pearson product-moment correlation was calculated for the average of the time series of the ROI (see Fig. 3A) to build a 200x200 connectivity matrix for each subject. Only positive correlation values in functional connectivity matrices were considered in this study.

We trained a single-layer neural network model on TensorFlow for classifying the Brain Imaging data. The parameters of the model were regularized by an elastic-net regularizer. The leave-one-out cross validation testing accuracy is around 61.17% that matches the state-of-the-art results [14, 28, 35] on this dataset. The logits of this network are used as model prediction scores, and

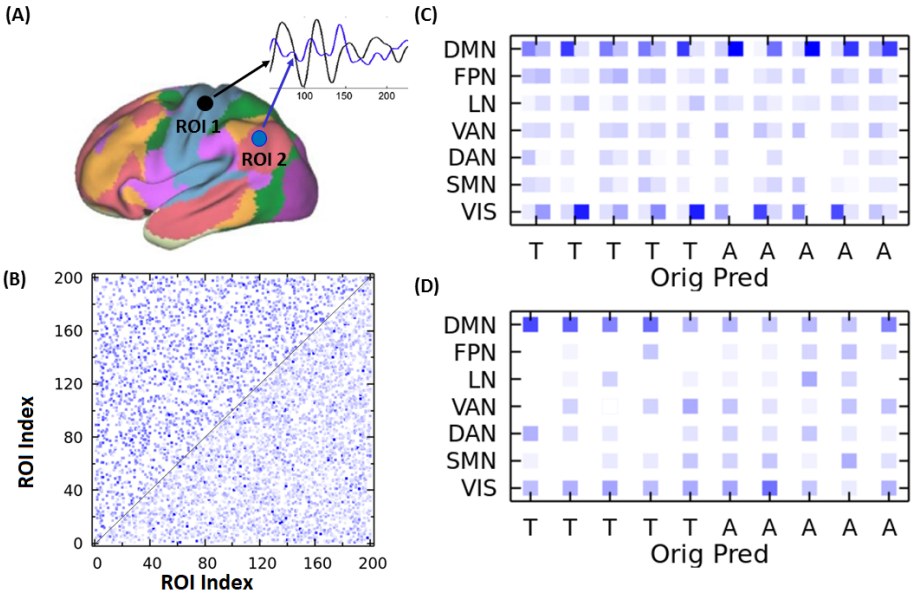

Figure 4: Comparison of CEM versus LRP on pre-processed resting-state brain fMRI connectivity data from the open-access ABIDE I database. (A) Seven networks of functionally coupled regions across the cerebral cortex, as obtained from the resting-state functional connectivity MRI [38]. The color scheme is as follows. Purple: Visual (VIS), blue: Somatomotor (SMN), green: Dorsal Attention (DAN), violet: Ventral Attention (VAN), cream; Limbic (LN), orange: Frontoparietal (FPN), and red: default mode (DMN). (B) CEM pertinent positives (PP) for the fMRI connectivity matrix of a CEM-classified autistic brain are shown in upper triangle, whereas the CEM pertinent negatives (PN) are shown in lower triangle. The color intensity represents strength of the PP and PN functional connections between regions of interest (ROI) in the brain. (C) PP and PN functional connections involving ROIs from individual brain functional network in CEM-classified autistic (denoted as A) and neurotypical (denoted as T) subjects. In each column, the left square represents the PP probability and right square represents the PN probability for an individual subject (T or A). Bolder the color higher the probability. (D) For LRP, we depict positive relevance of functional connections involving each of the seven functional network.

we set $\mathcal{X} = [0,1]^p$, $\mathcal{X}/\mathbf{x}_0 = [0,1]^p/\mathbf{x}_0$ and $\mathcal{X} \cap \mathbf{x}_0 = [0,1]^p \cap \mathbf{x}_0$ for any natural example $\mathbf{x}_0 \in \mathcal{X}$.

4.3.2 Results

With the help of domain experts, we evaluated the performance of CEM and LRP, which performed the best. LIME was challenging to use in this case, since the brain activity patterns are spread over the whole image and no reasonable segmentation of the images forming superpixels was achievable here.

Ten subjects were randomly chosen, of which five were classified as autistic and the rest as neuro-typical. Since the resting-state functional connectivity within and between large-scale brain functional networks [38] (see Fig. 3A) are often found to be altered in brain disorders including autism, we decided to compare the performance of CEM and LRP in terms of identifying those atypical patterns. Fig. 3B shows the strong pertinent positive (upper triangle) and pertinent negative (lower triangle) functional connections (FC) of a classified ASD subject produced by the CEM method. We further group these connections with respect to the associated brain network (Fig. 3C). Interestingly, in four out of five classified autistic subjects, pertinent positive FCs are mostly (with a probability > 0.26) associated with the visual network (VIS, shown in purple in Fig 3A). On the other hand, pertinent negative FCs in all five subjects classified as autistic preferably (with a probability > 0.42) involve the default mode network (DMN, red regions in Fig, 3A). This trend appears to be reversed in subjects classified as typical (Fig. 3C). In all five typical subjects, pertinent positive FCs involve DMN (with probability > 0.25), while the pertinent negative FCs correspond to VIS. Taken together, these results are consistent with earlier studies, suggesting atypical pattern of brain connectivity in autism [16]. The results obtained using CEM suggest under-connectivity involving DMN and over-connectivity in visual network, consistent with prior findings [16, 22]. LRP also identifies positively relevant FCs that mainly involve DMN regions in all five typical subjects. However, positively relevant FCs are found to be associated with the visual network in only 40% of autistic subjects. These findings imply superior performance of CEM compared to LRP in robust identification of pertinent positive information from brain functional connectome data of different populations.

To the best of our knowledge, this is the first-ever application of neural network models in classifying fMRI data from ABIDE I. The classification accuracy matches the best existing methods that automatically identify meaningful biomarkers of brain disorders. The extraction of pertinent positive and negative features by CEM can further help reduce error (false positives and false negatives) in such diagnoses.

5 Discussion

In the previous sections, we showed how our method can be effectively used to create meaningful explanations in different domains that are presumably easier to consume as well as more accurate. It's interesting that pertinent negatives play an essential role in many domains, where explanations are important. As such, it seems though that they are most useful when inputs in different classes are "close" to each other. For instance, they are more important when distinguishing a diagnosis of flu or pneumonia, rather than say a microwave from an airplane. If the inputs are extremely different then probably pertinent positives are sufficient to characterize the input, as there are likely to be many pertinent negatives, which will presumably overwhelm the user.

We believe that our explanation method CEM can be useful for other applications where the end goal may not be to just obtain explanations for the question, *why is input x classified into class y?*. Here are a few that we envision:

a) Model Selection: Consider two models that have the same (or similar) test performance. In such a scenario, we could use CEM to see which model provides better explanations. A model that provides better explanations may turn out to have better generalizability once deployed or may turn out to be more robust, both of which are critical in real applications. Robustness could be tested by applying state-of-the-art [7] adversarial attack methods and seeing if we need larger perturbations to create an adversary on average for the model with better explanations.

b) Contrastive Targeted Explanations: Rather than the above question, one could ask, *why is input x classified in class y, but not y'?*. In this case, we cannot directly apply CEM as it searches for the least perturbation to change classification to an arbitrary class. However, in such a case we can update our objectives where rather than doing a maximum over all other classes ($\max_{i \neq y} \text{Pred}(.)$), we could just insert the class of interest say y' ($\text{Pred}(.) = y'$) and obtain targeted contrastive explanations.

c) Model debugging: Our pertinent positives and negatives may be used to analyze biases of the model in terms of where and what kind of errors it is making often. This may help in interpreting the model as a whole. Ideally, it would be great if we could use this information and somehow improve the model with it not repeating such errors.

We plan to explore these directions in the future. Another idea to capture the data manifold may be to train a Generative Adversarial Network (GAN) [12] as opposed to an autoencoder. GANs have been used to generate natural adversaries [39] and can be used to smoothly transition between classes. In general though, good generation is, loosely speaking, a sufficient condition for ensuring that we have accurately captured the data manifold. This may be challenging in many applications and somewhat of an overkill for our purposes. Another thing that is unclear is how to efficiently navigate in the latent space given the add only and subtract only constraints that we have in the input space. It would be an interesting problem to translate these to the latent space. As

such, recent work [26] has shown that high quality images can be generated much more easily with autoencoders than with GANs. Nonetheless, this would be an interesting direction to explore especially for problem settings where high quality generators are already available.

In summary, we have provided a novel explanation method called CEM, which finds not only what should be minimally present in the input to justify its classification by black box classifiers such as neural networks, but also finds contrastive perturbations, in particular, additions, that should be necessarily absent to justify the classification. To the best of our knowledge this is the first explanation method that achieves this goal. We have validated the efficacy of our approach on multiple datasets from different domains, and shown the power of such explanations in terms of matching human intuition, thus making for more complete and well-rounded explanations.

References

[1] 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):e0130140, 2015.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[3] Dun & Bradstreet. Duns numbers. In *US Govt.* 2013. http://fedgov.dnb.com/webform.

[4] Industry & Security Bureau. Denied persons list. In *US Dept. of Commerce*, 2013. http://www.bis.doc.gov/index.php/policy-guidance/lists-of-parties-of-concern/denied-persons-list.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[6] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1721–1730, New York, NY, USA, 2015. ACM.

[7] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, 2018.

[8] R Cameron Craddock, G Andrew James, Paul E Holtzheimer, Xiaoping P Hu, and Helen S Mayberg. A whole brain fmri atlas generated via spatially

constrained spectral clustering. *Human brain mapping*, 33(8):1914–1928, 2012.

[9] Amit Dhurandhar, Vijay Iyengar, Ronny Luss, and Karthikeyan Shanmugam. Tip: Typifying the interpretability of procedures. *arXiv preprint arXiv:1706.02952*, 2017.

[10] Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659, 2014.

[11] Finale Doshi-Velez, Ryan Budish Mason Kortz, Chris Bavitz, David O'Brien Sam Gershman, Stuart Schieber, James Waldo, David Weinberger, and Alexandra Wood. Accountability of ai under the law: The role of explanation. *arXiv preprint arXiv:1711.01134*, 2017.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016.

[13] Karthik Gurumoorthy, Amit Dhurandhar, and Guillermo Cecchi. Protodash: Fast interpretable prototype selection. *arXiv preprint arXiv:1707.01212*, 2017.

[14] Anibal Sólon Heinsfeld, Alexandre Rosa Franco, R Cameron Craddock, Augusto Buchweitz, and Felipe Meneguzzi. Identification of autism spectrum disorder using deep learning and the abide dataset. *NeuroImage: Clinical*, 17:16–23, 2018.

[15] Amy Herman. Are you visually intelligent? what you dont see is as important as what you do see. *Medical Daily*, 2016.

[16] Jocelyn V Hull, Zachary J Jacokes, Carinna M Torgerson, Andrei Irimia, and John Darrell Van Horn. Resting-state functional connectivity in autism spectrum disorders: A review. *Frontiers in psychiatry*, 7:205, 2017.

[17] Tsuyoshi Id and Amit Dhurandhar. Supervised item response models for informative prediction. *Knowl. Inf. Syst.*, 51(1):235–257, April 2017.

[18] Transparency Intl. Corruption perceptions index. 2013. http://www.transparency.org/research/cpi/overview.

[19] Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *In Advances of Neural Inf. Proc. Systems*, 2016.

[20] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. The lrp toolbox for artificial neural networks. *Journal of Machine Learning Research*, 17(114):1–5, 2016.

[21] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

[22] Adam Liska, Hongyuan You, and Payel Das. Relationship between static and dynamic brain functional connectivity in autism spectrum disorders. *presented at the ISMRM*, in Honolulu, 2017.

[23] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

[24] Ali Mousavi, Gautam Dasarathy, and Richard G. Baraniuk. Deepcodec: Adaptive sensing and recovery via deep convolutional neural networks. *arXiv preprint arXiv:1707.03386*, 2017.

[25] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[26] Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

[28] Jared A Nielsen, Brandon A Zielinski, P Thomas Fletcher, Andrew L Alexander, Nicholas Lange, Erin D Bigler, Janet E Lainhart, and Jeffrey S Anderson. Multisite functional connectivity mri classification of autism: Abide results. *Frontiers in human neuroscience*, 7:599, 2013.

[29] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you? explaining the predictions of any classifier. In *ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining*, 2016.

[30] Su-In Lee Scott Lundberg. Unified framework for interpretable methods. In *In Advances of Neural Inf. Proc. Systems*, 2017.

[31] 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. *See https://arxiv. org/abs/1610.02391 v3*, 2016.

[32] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[33] Guolong Su, Dennis Wei, Kush Varshney, and Dmitry Malioutov. Interpretable two-level boolean rule learning for classification. In *https://arxiv.org/abs/1606.05798*, 2016.

[34] Award Management System. Excluded parties list. In *US Govt.*, 2013. https://www.sam.gov/portal/public/SAM/.

[35] Ravi Tejwani, Adam Liska, Hongyuan You, Jenna Reinen, and Payel Das. Autism classification using brain functional connectivity dynamics and machine learning. *NIPS workshop BigNeuro*, 2017.

[36] Fulton Wang and Cynthia Rudin. Falling rule lists. In *In AISTATS*, 2015.

[37] Yoshua Bengio Yann LeCun, Leon Bottou and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278 2324, 1998.

[38] BT Thomas Yeo, Fenna M Krienen, Jorge Sepulcre, Mert R Sabuncu, Danial Lashkari, Marisa Hollinshead, Joshua L Roffman, Jordan W Smoller, Lilla Zöllei, Jonathan R Polimeni, et al. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of neurophysiology*, 106(3):1125–1165, 2011.

[39] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.

[40] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.