Evaluation of CNN Performance in Semantically Relevant Latent Spaces

Jeroen van Doorenmalen^(✉) and Vlado Menkovski^(✉)

Eindhoven University of Technology, Eindhoven, The Netherlands
j.v.doorenmalen@student.tue.nl, v.menkovski@tue.nl

Abstract. We examine deep neural network (DNN) performance and behavior using contrasting explanations generated from a semantically relevant latent space. We develop a semantically relevant latent space by training a variational autoencoder (VAE) augmented by a metric learning loss on the latent space. The properties of the VAE provide for a smooth latent space supported by a simple density and the metric learning term organizes the space in a semantically relevant way with respect to the target classes. In this space we can both linearly separate the classes and generate meaningful interpolation of contrasting data points across decision boundaries. This allows us to examine the DNN model beyond its performance on a test set for potential biases and its sensitivity to perturbations of individual factors disentangled in the latent space.

Keywords: Deep learning · VAE · Metric learning · Interpretability · Explanation

1 Introduction

Advances in machine learning and deep learning have had a profound impact on many tasks involving high dimensional data such as object recognition and behavior monitoring. The domain of Computer Vision especially has been witnessing a great growth in bridging the gap between the capabilities of humans and machines. This field tries to enable machines to view the world as humans do, perceive it similar and even use the knowledge for a multitude of tasks such as Image & Video Recognition, Image Analysis and Classification, Media Recreation, recommender systems, etc. And, has since been implemented in high-level domains like COMPAS [8], healthcare [3] and politics [17]. However, as black-box models inner workings are still hardly understood, can lead to dangerous situations [3], such as racial bias [8], gender inequality [1].

The need for confidence, certainty, trust and explanations when using supervised black-box models is substantial in domains with high responsibility. This paper provides an approach towards better understanding of a model's predictions by investigating its behavior on semantically relevant (contrastive) explanations. The build a semantically relevant latent space we need a smooth space

that corresponds well with the generating factors of the data (i.e. regions well-supported by the associated density should correspond to realistic data points) and with a distance metric that conveys semantic information about the target task. The vanilla VAE without any extra constraints is insufficient as it does not necessarily deliver a distance metric that corresponds to the semantics of the target class assignment (in our task). Our target is to develop semantically relevant decision boundaries in the latent space, which we can use to examine our target classification model. Therefore, we propose to use a weakly-supervised VAE that uses a combination of metric learning and VAE disentanglement to create a semantically relevant, smooth and well separated space. And, we show that we can use this VAE and semantically relevant latent space can be used for various interpretability/explainability tasks, such as validate predictions made by the CNN, generate (contrastive) explanations when predictions are odd and being able to detect bias. The approach we propose for these tasks is more specifically explained using Fig. 1.

Fig. 1. The diagnostics approach to validate and understand the behavior of the CNN. (1) extra constraints, loss functions are applied during training of the VAE in order to create semantically relevant latent spaces. The generative model captures the essential semantics within the data and is used by (2) A linear Support Vector Machine. The linear SVM is trained on top of the latent space to classify input on semantics rather than the direct mapping from input data X and labels Y . If the SVM and CNN do not agree on a prediction then (3) we traverse the latent space in order to generate and capture semantically relevant synthetic images, tested against the CNN, in order to check what elements have to change in order to change its prediction from a to b , where a and b are different classes.

In this paper, the key contributions are: (1) an approach that can be used in order to validate and check predictions made by a CNN by utilizing a weakly-supervised generative model that is trained to create semantically relevant latent spaces. (2) The semantically relevant latent spaces are then used in order to train a linear support vector machine to capture decision rules that define a class assignment. The SVM is then used to check predictions based on semantics

rather than the direct mapping of the CNN. (3) if there is a misalignment in the predictions (i.e. the CNN and SVM do not agree) then we posit the top k best candidates (classes) and for these candidates traverse the latent spaces in order to generate semantically relevant (contrastive) explanations by utilizing the decision boundaries of the SVM.

To conclude, This paper posits a method that allows for the validation of CNN performance by comparing it against the linear classifier that is based on semantics and provides a framework that generates explanations when the classifiers do not agree. The explanations are provided qualitatively to an expert within the field. This explanation encompasses the original image, reconstructed images and the path towards its most probable answers. Additionally, it shows the minimal difference that makes the classifiers change its prediction to one of the most probable answers. The expert can then check these results to make a quick assessment to which class the image actually belongs to. Additionally, the framework provides the ability to further investigate the model mathematically using the linear classifier as a proxy model.

2 Related Work

Interest in interpretability and explainability studies has significantly grown since the inception of “Right to Explanation” [20] and ethicality studies into the behavior of machine learning models [1, 3, 8, 17]. As a result, developers of AI are promoted and required, amongst others, to create algorithms that are transparent, non-discriminatory, robust and safe. Interpretability is most commonly used as an umbrella term and stands for providing insight into the behavior and thought processes behind machine-learning algorithms and many other terms are used for this phenomenon, such as, Interpretable AI, Explainable machine learning, causality, safe AI, computational social science, etc. [5]. We posit our research as an interpretability study, but it does not necessarily mean that other interpretability studies are directly closely related to this work.

There have been many approaches that all work towards the goal of understanding black-box models: Linear Proxy Models: Lime [18] are approaches that locally approximate complex models using linear fits, Decision trees and Rule extraction methods, such as deepred [21] are also considered highly explainable, but quickly become intractable as complexity increases and salience mapping [19] that provide visual information as to which part of an image is most likely used in its prediction, however, it has been demonstrated to be unreliable if not strongly conditioned [10]. Additionally, another approach to interpretability is explaining the role of each part within a black-box models such as the role of a layer or individual neurons [2] or representation vectors within the activation space [9].

Most of the approaches stated above assume that there has to be a trade-off between model performance and explainability. Additionally, as the current interpretable methods for black-box models are still insufficient and approximated can cause more harm than good when communicated as a method that

solves all problems. A lot of the interpretability methods do not take into account the actual needs that stakeholders require [13]. Or, fail to take into account the vast research into explanations or interpretability of the field of psychology [14] and social sciences [15]. The “Explanation in Artificial Intelligence” study by Miller [15] describes the current state of interpretable and explainable algorithms, how most of the techniques currently fail to capture the essence of an explanation and how to improve: an interpretability or explainability method should at least include, but is not limited to, a non-disputable textual- and/or mathematical- and/or visual explanation that is selective, social and depending on the proof, contrastive.

For this reason, our approach focuses on providing selective (contrastive) explanations that combines visual aspects as well as the ability to further investigate the model mathematically using a proxy model that does not impact the CNN directly. Usually, generative models such as the Variational Autoencoders (VAE) [11] and Generative Adversarial Networks (GAN)s are unsupervised and used in order to sample and generate images from a latent space, provided by training the generative network. However, we posit to use a weakly-supervised generative network in order to impose (discriminative) structure in addition to variational inference to the latent space of said model using metric learning [6].

This approach and method is therefore most related to the interpretability area of sub-sampling proxy generative models to answer questions about a discriminative black box model. The two closest studies that attempt similar research is a preprint of CDeepEx [4] by Amir Feghahati et al. and xGEMs [7] by Joshi et al. Both cDeepEx and xGEMS propose the use of a proxy generative model in order to explain the behavior of a black-box model, primarily using generative adversarial networks (GANs). The xGems paper presents a framework to characterize and explaining binary classification models by generating manifold guided examples using a generative model. The behavior of the black box model is summarized by quantitatively perturbing data samples along the manifold. And, xGEMS detects and quantifies bias during model training to understand how bias affects black box models. The xGEMS approach is similar to our approach as in using a generative model in order to explain a black box model. Similarly, the cDeepEx paper posits their work as generating contrastive explanations using a proxy generative model. The generated explanations focus on answering the question “why a and not b ?” with GANs, where a is the class of an input example I and b is a chosen class to which to capture the differences.

However, both of these papers do not state that in a multi-class (discriminative) classification problem if the generative models’ latent space is not smooth, well separated and semantically relevant then unexpected behavior can happen. For instance, when traversing the latent space it is possible to can pass from a to any number of classes before reaching class b because the space is not well separated and smooth. This will create ineffective explanations, as depending on how they generate explanations will give information on ‘why class a and not b using properties of c ’. An exact geodesic path along the manifold would require great effort, especially in high dimensions. Also, our approach is different in the

fact that we utilize a weakly-supervised generative model as well as an extra linear classifier on top of the latent space to provide us with extra information on the data and the latent space. Some approaches we take, however, are very similar, such as using a generative model as a proxy to explain a black-box model as well as sub-sampling the latent space to probe the behavior of a black-box model and generate explanations using the predictions.

3 Methodology

This paper posits its methodology as a way to explain and validate decisions made by a CNN. The predictions made by the CNN are validated and explained utilizing the properties of a weakly-supervised proxy generative model, more specifically, a triplet-vae. There are three main factors that contribute to the validation and explanation of the CNN. First, a triplet-vae is trained in order to provide a semantically relevant and well separated latent space. Second, this latent space is then used to train an interpretable linear support vector machine and is used to validate decisions by the CNN by comparison. Third, when a CNN decision is misaligned with the decision boundaries in the latent space, we generate explanations through stating the K most probable answers as well as provide a qualitative explanation to validate the top K most probable answers. Each of these factors respectively refer to the number stated in Fig. 1 as well as link to each section: (1) triplet-vae Sect. 3.1, (2) CNN Decision Validation, Sect. 3.2, (3) Generating (contrastive) Explanations, Sect. 3.3.

3.1 Semantically Relevant Latent Space

Typically, a triplet network consists of three instances of a neural network that share parameters. These three instances are separately fed differences types of input: an anchor, positive sample and negative sample. These are then used to learn useful representations by distance comparisons. We propose to incorporate this notion of a triplet network to semantically structure and separate the latent space of the VAE using the available input and labels. A triplet VAE consists of three instances of the encoder with shared parameters that are each fed pre-computed triplets: an anchor, positive sample and negative sample; x_a , x_p and x_n . The anchor x_a and positive sample x_p are of the same class but not the same image, whereas negative sample x_n is from a different class. In each iteration of training, the input triplet is fed to the encoder network to get their mean latent embedding: $\mathcal{F}(x_a)^\mu = z_a^\mu$, $\mathcal{F}(x_p)^\mu = z_p^\mu$, $\mathcal{F}(x_n)^\mu = z_n^\mu$. These are then used to compute a similarity loss function as to induce loss when a negative sample z_n^μ is closer to z_a^μ than z_p^μ distance-wise. i.e. $\delta_{ap}(z_a^\mu, z_p^\mu) = \|z_a^\mu - z_p^\mu\|$ and $\delta_{an}(z_a^\mu, z_n^\mu) = \|z_a^\mu - z_n^\mu\|$ and, provides us with three possible situations: $\delta_{ap} > \delta_{an}$, $\delta_{ap} < \delta_{an}$ and $\delta_{ap} = \delta_{an}$ [6].

We wish to find an embedding where samples of a certain class lie close to each other in the latent space of the VAE. For this reason, we wish to add loss the algorithm when we arrive in the situation where $\delta_{ap} > \delta_{an}$. In other words,

Fig. 2. Given an input image I we check the prediction of the CNN as well as the SVM. If both classifiers predict the same class, we return the predicted class. In contrast, if the classifiers do not predict the same class, we propose to return the top k most probable answers as well as an explanation why those classes are the most probable.

we wish to push x_n further away, such that we ultimately arrive in the situation where $\delta_{ap} < \delta_{an}$ or $\delta_{ap} = \delta_{an}$ with some margin ϕ . As such we arrive at the triplet loss function that we'll use in addition to the KL divergence and reconstruction loss within the VAE: $L(z_a^\mu, z_p^\mu, z_n^\mu) = \alpha * \text{argmax}\{\|z_a^\mu - z_p^\mu\| - \|z_a^\mu - z_n^\mu\| + \phi, 0\}$. Where ϕ will provide leeway when $\delta_{ap} = \delta_{an}$ and push the negative sample away even when the distances are equal.

We have an already present CNN which we would like to validate, and is trained by input data $X : x_1 \dots x_n$ and labels $Y : y_1 \dots y_n$ where each y_i states the true class of x_i . We then use the same X and Y to train the triplet-VAE. (1) First, we compute triplets of the form x_a, x_p, x_n from the input data X and labels Y which are then used to train the triplet VAE. A typical VAE consists of an $\mathcal{F}(x) = \text{Encoder}(x) \sim q(z|x)$ which compresses the data into a latent space Z , a $\mathcal{G}(z) = \text{Decoder}(z) \sim p(x|z)$ which reconstructs the data given the latent space Z and a prior $p(z)$, in our case a gaussian $\mathcal{N}(0,1)$, imposed on the model. In order for the VAE to train a latent space similar to its prior and be able to reconstruct images it is trained by minimizing the Evidence Lower Bound (ELBO). $ELBO = -\mathbb{E}_{z \sim \mathcal{Q}(z|X)}[\log P(x|z)] + \mathcal{KL}[\mathcal{Q}(z|X)||P(z)]$ This can be explained as the reconstruction loss or expected negative loglikelihood: $-\mathbb{E}_{z \sim \mathcal{Q}(z|X)}[\log P(x|z)]$ and the KL divergence loss $\mathcal{KL}[\mathcal{Q}(z|X)||P(z)]$, to which we add the triplet loss:

$$\mathcal{L}(z_a^\mu, z_p^\mu, z_n^\mu) = \alpha * \text{argmax}\{\|z_a^\mu - z_p^\mu\| - \|z_a^\mu - z_n^\mu\| + \phi, 0\}$$

This compound loss semi-forces the latent space of the VAE to be well separated due to the triplet loss, disentangled due to the KL divergence loss combined with β scalar, and provides a means of (reasonably) reconstructing images by the reconstruction loss. And, thus results in the following loss function for training the VAE:

$$\text{loss} = -\mathbb{E}_{z \sim \mathcal{Q}(z|X)}[\log P(x|z)] + \beta * \mathcal{KL}[\mathcal{Q}(z|X)||P(z)] + \mathcal{L}(z_a^\mu, z_p^\mu, z_n^\mu).$$

3.2 Decision Validation

Afterwards, given a semantically relevant latent space we can use it for step two and three as indicated in Fig. 1. (2) Second step - CNN Decision Validation, we train an additional classifier on top of the triplet-VAE latent space, specifically z^μ . We train the linear Support Vector Machine using Z^μ s as input data and Y as labels where $[Z^\mu, Z^\sigma] = \mathcal{F}(\mathcal{X})$. The goal of the linear support vector machine is two-fold. It provides a means of validating each prediction made by the CNN by using the encoder and the linear classifier. i.e. given an input example I , we have $\mathcal{C}(I) = \hat{y}_{\mathcal{C}(I)}$ and $\mathcal{S}(\mathcal{F}(I)^\mu) = \hat{y}_{\mathcal{S}(I)}$, and compare them against each other $\hat{y}_{\mathcal{C}(I)} = \hat{y}_{\mathcal{S}(I)}$. And, as the linear classifier is a simpler model than the highly complex CNN it will function as the ground-truth base for the predictions that are made. As such, we arrive at two possible cases:

$$\text{Comparison}(I) = \begin{cases} \text{Positive} & \text{if } (\hat{y}_{\mathcal{C}(I)} = \hat{y}_{\mathcal{S}(I)}) \\ \text{Negative} & \text{if } (\hat{y}_{\mathcal{C}(I)} \neq \hat{y}_{\mathcal{S}(I)}) \end{cases} \quad (1)$$

First, If both classifiers agree then we arrive at an optimal state, meaning that the prediction is based on semantics and the direct mapping found by the CNN. In this way, we can say with high confidence that the prediction is correct. In the second case, if the classifiers do not agree, three cases can occur: the SVM is correct and the CNN is incorrect, the SVM is incorrect and the CNN is correct, or both the SVM and the CNN is incorrect. In each of these cases we can suggest a most probable answer as well as a selective (contrastive) explanation indicated as step 3 of the framework as explained in Fig. 2.

3.3 Generating (contrastive) Explanations

An explanation consists of (1) the most probable answers and (2) a qualitative investigation of latent traversal towards the most probable answers. The most probable answer is presented by the averaged sum rule [12] over the predicted probabilities per class for both the CNN and SVM and selecting the top K answers, where K can be appropriately selected. Additionally, originally an SVM does not return a probabilistic answer, however, applying Platt's [16] method we apply an additional sigmoid function to map the SVM outputs into probabilities. These top k answers are then used in order to present and generate selected contrastive explanations.

The top K predictions or classes will be used in order to traverse and sub-sample the latent space from the initial representation or Z_I^μ location towards another class. We can find a path by finding the closest point within the latent space such that the decision boundary is crossed and the SVM predicts the target class. Alternatively we could use the closest data point in the latent space that adheres to the training set $\text{argmin } \mathcal{F}(x_i)^\mu - Z_I^\mu$ for every $x_i \in X$. Traversing and sub-sampling the latent space will change the semantics minimally to change the class prediction. We capture the minimal change needed in order to change both the SVM and CNN prediction to the target class. This information is then

Fig. 3. Generating (contrastive) explanations consist of several steps: First, given an input image I in question and the K top most probable answer. K denotes training data X for class k labeled with $y = k$. We feed both I and K through the encoder $\mathcal{F}(X)$ to receive their respective semantic location in the latent space. We then find the closest training point that belongs to the target class k and find the vector $\vec{v}$; the direction of that point. Afterwards, uniformly sample ϵ data points along this vector $\vec{v}$, where j iterates over $0 \dots j \dots \epsilon$ and is denoted as Z_v^μ . Z_v^μ is then used to check these against the SVM and use them to generate images $X_{Z_v^\mu}$ using the decoder $\mathcal{G}(Z_v^\mu)$. The generated images are then fed to the CNN to make a prediction and as the images will semantically change along the vector the prediction will change as well. Afterwards, we can compare the predictions from both the CNN and SVM. Subsequently, we use the first moment where both predictions are equal to target class k , denoted as moment l for generating an explanation - minimal semantic difference necessary to be equal to the target class, ΔU_l .

presented to the domain expert for verification and answers the following question: The most probable answer is a because the input image I is semantically closest to the following features, where the features are presented qualitatively. The explanations are generated as follows: see Fig. 3.

The decision boundaries around the clusters within the latent space are fitted by the SVM and can be used to answer questions of the form ‘why a and not b ?’. If $\hat{y}_{C(I)}$ and $\hat{y}_{S(I)}$ do not predict the same class, then, we assume that $\hat{y}_{S(I)}$ is correct. We then use the find a path, indicated by $\vec{v}$ from $\hat{y}_{S(I)}$ to $\hat{y}_{C(I)}$, Z_I^μ to the target class. This can be done by calculating a vector orthogonal to the hyper-plane fitted by the SVM towards the target class. Alternatively, we can find the closest $z^\mu \in Z^\mu$ that satisfies $\hat{y}_{S(z^\mu)} = \hat{y}_{C(z^\mu)}$ that are not the same as the initial prediction $\hat{y}_{C(I)}$. This means that $\vec{v}$ is the vector from I to the closest data point of the target class, with respect to Euclidean distance.

We then uniformly sample points along vector $\vec{v}$ and check them against the SVM as well as the CNN. The sampled points can directly be fed to the SVM to get a prediction $\hat{y}(f(v_i))$ for every $v_i \in V$. Similarly, we can get predictions of the CNN by transforming the images using the decoder $\mathcal{D}$. The images are then fed to

the CNN to get a prediction $\hat{y}(\mathcal{C}(\mathcal{D}(v_i)))$ for every $v_i \in V$. The predictions of both classifiers will change as the images start looking more and more like the target class as generative factors change along the vector. If we capture the changes that make the change happen, we can show the minimal difference required in order to change the prediction of the CNN. In this way we can generate contrastive examples: For the top ‘close’ class that is not $\hat{y}_I$ we answer the question: ‘why $\hat{y}_I$ and not the other semantically close class?’. Hence, we find the answer to the question “why a and not b ?”, as the answer is the shortest approximate changes between the two classes that make the CNN change its prediction. As a result, we have found a way to validate the inner workings of the CNN. If there are doubts about a prediction it can be investigated and checked.

4 Results

In this paper we show experimental results on MNIST by generating (contrastive) explanations to provide extra information to predictions made by the CNN and evaluate its performance. The creation of these explanations requires a semantically relevant and well separated latent space. Therefore, we first show the difference between the latent space of the vanilla VAE and the triplet-VAE and its effects on training a linear classifier on top of the latent space. The Figs. 4 and 5 show a tSNE visualization of the separation of classes within the latent space. Not surprisingly the triplet-VAE separated the data in a far more semantically relevant way and this is also reflected with respect to the accuracy of training a linear model on the data.

Fig. 4. Visualization of a two-dimensional latent space of a vanilla VAE on MNIST

Fig. 5. Visualization of a two-dimensional latent space of a $\mathcal{T}$ -VAE on MNIST

Second, the percentages show as to know how much both classifiers agree by showing the percentage per possible case, as shown in Table 1. Not surprisingly case four happens more often than case three and can mean two things, our latent space is too simple to capture the full complexity of the class assignment and the CNN is not constraint by extra loss functions. However, in three of the four cases where $Y_S \neq Y_C$ we can explain the most probable predictions and provide a generated (contrastive) explanation. The only case we cannot check or know about is case two, where both Y_S and Y_C predict the same class but is wrong. The only way to capture this behavior is by explaining every single decision by generating explanations for everything. Nevertheless, as an example for generating explanations we use an example: 6783 (case 5) as shown in Fig. 6.

Generating explanations consists of three parts: First, we propose the top K probable answers: for this example the true label is 1, the most probable answers are 6, 8 and then 1 with averaged probabilities 0.512332, 0.3382, 0.1150. Second, Then for those most probable target classes, 6, 8, 1 we traverse the latent space from the initial location Z_I^μ to the closest point of that class, denoted as $v \in$ that is predicted correctly i.e. the SVM and CNN agree. Figure 7 shows the generated images from the uniformly sampled data points along vectors $v_k \in V$ where $k \in K$ stand for 6, 8, 1 in this case. The figures show which changes happen when traversing the latent space and at which points both the SVM and the CNN agree with respect to their decision.

For the traversal from Z_I^μ to class 6 it can be seen that rather quickly both classifiers agree and only minimal changes are required to change the predictions. Third, for such an occurrence we can further zoom in on what is happening and what really makes that the most probable answer. Figure 6 shows these minimal changes required to change its prediction as well as the transformed image on which the classifiers agree. The first row shows the original image, positive changes, negative changes and the changes combined. The second row shows the reconstructed image and the reconstructed images with the positive changes, negative changes and positive and negative changes respectively. In this way, for each probable answer it shows its closest representative and the changes required to be part of that class.

Table 1. This table shows the percentages of agreement with respect to all possible cases.

Case	Percentage
(1) $\hat{Y}_S = \hat{Y}_C = Y$	0.9586
(2) $\hat{Y}_S = \hat{Y}_C \neq Y$	0.003
(3) $(\hat{Y}_S = Y) \neq \hat{Y}_C$	0.0086
(4) $\hat{Y}_S \neq (\hat{Y}_C = Y)$	0.0314
(5) $\hat{Y}_S \neq \hat{Y}_C \neq Y$	0.0044

Fig. 6. Once the SVM and the CNN both predict the target class we capture the minimal changes that are necessary to change their predictions

Fig. 7. Per top k probable answers we traverse and sample the latent space to generate images that can be used to test the behavior of the CNN. The red line indicates the moment where both the SVM and the CNN predict the target class (Color figure online)

5 Conclusion

This paper examines deep neural network’s behaviour and performance by utilizing a weakly-supervised generative model as a proxy. The weakly-supervised generative model aims to uncover the generative factors underlying the data and separate abstract classes by applying metric learning. The proxy’s goal is three-fold: the semantically meaningful space will be the base for a linear support vector machine; The model’s generative capabilities will be used to generate images that can be probed against the black box in question; the latent space is traversed and sampled from an anchor I to another class k in order to find the minimal important difference that changes both classifier’s predictions. The goal of the framework is to be sure of the predictions made by the black box by better understanding the behaviour of the CNN by simulating questions of the form ‘Why a and not b ?’ where a and b are different classes.

We examine deep neural network (DNN) performance and behaviour using contrasting explanations generated from a semantically relevant latent space. The results show that each of the above goals can be achieved and the framework performs as expected. We develop a semantically relevant latent space by training an variational autoencoder (VAE) augmented by a metric learning loss on the latent space. The properties of the VAE provide for a smooth latent space supported by a simple density and the metric learning term organizes the space in a semantically relevant way with respect to the target classes. In this space we can both linearly separate the classes and generate relevant interpolation of contrasting data points across decision boundaries and find the minimal important difference that changes the classifier’s predictions. This allows us to examine the DNN model beyond its performance on a test set for potential biases and its sensitivity to perturbations of individual factors in the latent space.

References

1. Buolamwini, J., Gebru, T.: Gender shades: intersectional accuracy disparities in commercial gender classification. In: Friedler, S.A., Wilson, C. (eds.) *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. *Proceedings of Machine Learning Research*, vol. 81, pp. 77–91. PMLR, New York, February 2018. <http://proceedings.mlr.press/v81/buolamwini18a.html>
2. Carter, S., Armstrong, Z., Schubert, L., Johnson, I., Olah, C.: Activation atlas. *Distill* (2019). <https://doi.org/10.23915/distill.00015>, <https://distill.pub/2019/activation-atlas>
3. Challen, R., Denny, J., Pitt, M., Gompels, L., Edwards, T., Tsaneva-Atanasova, K.: Artificial intelligence, bias and clinical safety. *BMJ Qual. Saf.* **28**(3), 231–237 (2019). <https://doi.org/10.1136/bmjqs-2018-008370>. <https://qualitysafety.bmj.com/content/28/3/231>
4. Feghahati, A., Shelton, C.R., Pazzani, M.J., Tang, K.: CDeepEx: contrastive deep explanations (2019). <https://openreview.net/forum?id=HyNmRiCqtm>
5. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: an approach to evaluating interpretability of machine learning. *CoRR abs/1806.00069* (2018). <http://arxiv.org/abs/1806.00069>
6. Ishfaq, H., Hoogi, A., Rubin, D.: TVAE: triplet-based variational autoencoder using metric learning (2018)
7. Joshi, S., Koyejo, O., Kim, B., Ghosh, J.: xGEMs: generating exemplars to explain black-box models. *CoRR abs/1806.08867* (2018). <http://arxiv.org/abs/1806.08867>
8. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias, May 2016. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>
9. Kim, B., et al.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 2668–2677, July 2018
10. Kindermans, P.J., et al.: The (un)reliability of saliency methods. *CoRR abs/1711.00867* (2017). <http://dblp.uni-trier.de/db/journals/corr/corr1711.html#abs-1711-00867>
11. Kingma, D.P., Welling, M.: An introduction to variational autoencoders. *CoRR abs/1906.02691* (2019). <http://arxiv.org/abs/1906.02691>
12. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(3), 226–239 (1998). <https://doi.org/10.1109/34.667881>
13. Lipton, Z.C.: The doctor just won’t accept that! In: *NIPS Proceedings 2017*, no. 24, pp. 1–3, November 2017. <https://arxiv.org/pdf/1711.08037.pdf>
14. Lombrozo, T.: Explanation and abductive inference. In: *Oxford Handbook of Thinking and Reasoning*, pp. 260–276 (2012). <https://doi.org/10.1093/oxfordhdb/9780199734689.013.0014>
15. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *CoRR abs/1706.07269* (2017). <http://arxiv.org/abs/1706.07269>
16. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press (1999)
17. Prakash, A.: Ai-politicians: a revolution in politics, August 2018. <https://medium.com/politics-ai/ai-politicians-a-revolution-in-politics-11a7e4ce90b0>

18. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you?: explaining the predictions of any classifier. CoRR abs/1602.04938 (2016). <http://arxiv.org/abs/1602.04938>
19. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-CAM: why did you say that? Visual explanations from deep networks via gradient-based localization. CoRR abs/1610.02391 (2016). <http://arxiv.org/abs/1610.02391>
20. European Union: Official journal of the European union: Regulations (2016). <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN>
21. Zilke, J.R., Loza Mencía, E., Janssen, F.: DeepRED – rule extraction from deep neural networks. In: Calders, T., Ceci, M., Malerba, D. (eds.) DS 2016. LNCS (LNAI), vol. 9956, pp. 457–473. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46307-0_29

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.