Interpretability of Deep Learning Models: A Survey of Results

Supriyo Chakraborty*, Richard Tomsett§, Ramya Raghavendra*, Daniel Harborne†, Moustafa Alzantot‡,
Federico Cerutti†, Mani Srivastava‡, Alun Preece†, Simon Julier††, Raghuveer M. Rao¶,
Troy D. Kelley¶, Dave Braines§, Murat Sensoy‖, Christopher J. Willis**, Prudhvi Gurram¶
*IBM T. J. Watson Research Center, †Crime and Security Research Institute, Cardiff University, ‡ UCLA, § IBM UK,
¶ Army Research Lab, Adelphi, ‖ Ozyegin University, ** BAE Systems AI Labs †† University College London

***Abstract*—Deep neural networks have achieved near-human accuracy levels in various types of classification and prediction tasks including images, text, speech, and video data. However, the networks continue to be treated mostly as black-box function approximators, mapping a given input to a classification output. The next step in this human-machine evolutionary process – incorporating these networks into mission critical processes such as medical diagnosis, planning and control – requires a level of trust association with the machine output.**

Typically, statistical metrics are used to quantify the uncertainty of an output. However, the notion of trust also depends on the visibility that a human has into the working of the machine. In other words, the neural network should provide human-understandable justifications for its output leading to insights about the inner workings. We call such models as interpretable deep networks.

Interpretability is not a monolithic notion. In fact, the subjectivity of an interpretation, due to different levels of human understanding, implies that there must be a multitude of dimensions that together constitute interpretability. In addition, the interpretation itself can be provided either in terms of the low-level network parameters, or in terms of input features used by the model. In this paper, we outline some of the dimensions that are useful for model interpretability, and categorize prior work along those dimensions. In the process, we perform a gap analysis of what needs to be done to improve model interpretability.

I. Introduction

Advances in machine learning and deep learning have had a profound impact on many "low-level" tasks such as object recognition and behaviour monitoring. Recently, researchers have begun to explore how these approaches can be used in "high-level" domains such as healthcare, criminal justice system, finance, and military decision making [1]. As the importance of the decisions aided using machine learning increases, it becomes more important for users to be able to suitably weight the assistance provided by such systems. A key property is *interpretability* — users should have the ability to *understand* and *reason* about the model output. However, despite several years of research effort, progress in this area remains limited [2]. For example, multi-layer neural networks, in spite of the their tremendous success in achieving near-human accuracy levels in certain prediction and classification tasks [3], operate as *black boxes*, and offer little to no explanation/visibility into why specific features are selected over others during training, or how the correlations in the training data are represented in the choice of the features, or why a specific pathway in the network (e.g., transforming raw data to classification output) is selected over others.

While deep learning based models are motivated by neuroscientific advancements in the understanding of the working of the human brain, a critical distinction, that has often been made between the two, is attributed to the human ability to "think" [4]. Informally, it is this ability to *think*, that allows humans to not only make a prediction, but also *justify* or *rationalize* it through a series of logically consistent and *understandable* choices leading up to the prediction. This justification, in turn, enables the decision maker to implicitly or explicitly associate a measure of *confidence* to the prediction aiding the decision making process. The counterpart to the human thought process in deep learning models is often referred to as *interpretability* [2].

One may argue that the above justification should be in terms of the low-level machine parameters and their sequential updates due to a learning algorithm. However, a closer inspection of even the human thought process reveals that we do not actually interpret the working of our brain in terms of its low-level parameters. We do not justify our predictions based on the learning algorithm used by the brain or the way it chooses to represent information (model parameters). Instead, it is typical to provide justifications, more often than not in a post-hoc setting, using prior information that can correlate model response with physical observations. This implies that the notion of interpretability is not restricted to model parameters alone but can be defined at multiple levels such as: model parameters and learning algorithms, or functionality of the model, or a combination of both.

In fact, as observed in [2], the notion of interpretability is not even a monolithic concept but reflects several different dimensions, which are summarized below:

- **Model Transparency:** This is defined in terms of three parameters: (i) *simulatability* – whether a human can use the input data together with the model to reproduce every calculation step necessary to make the prediction. This allows the human to understand the changes in the model parameters caused by the training data; (ii) *decomposability* – whether there is an intuitive explanation for all the model parameters; and finally (iii) *algorithmic transparency* – which is essentially an ability to explain the working of the learning algorithm. For example, the

choice of a hyperplane in the Support Vector Machine (SVM) can be explained in terms of the marginal points and the decision boundary. However, for a deep neural network, the non-linearities added into the features at each layer makes it difficult to explain the features being used for the output.

- **Model Functionality:** This is defined in terms of (i) *textual description* – providing a semantically meaningful description of the model output. To do so, one might use a composition of models, one for prediction and another one to generate a textual explanation; (ii) *visualization* – another common means of explaining the working of a model is through visualization of the parameters. One popular approach to visualize high-dimensional distributed representations is using the t-SNE mechanism [5]; and finally (iii) *local explanation* – where instead of explaining the entire mapping of a model, local changes introduced by a specific input vector for a given output class is computed. For example, in neural networks, the gradient of the output can be used to identify specific weights and the local changes that are influenced by the input vector.

In this paper, our primary contribution is a categorization of prior work on machine interpretability based on the above dimensions. We then provide a brief exposition of a coalition setting in which we want to train an interpretable deep neural network and conclude by identifying challenges that are unique to this setting and their influence on model interpretability.

II. Summary of prior art

In this section, we describe recent work on improving the interpretability of deep learning models. We classify each work according to the dimensions of interpretability introduced in the previous section, outlined by [2]. Note that this review is not fully comprehensive, but represents a survey of methods and results that we consider particularly pertinent to our future research goals in deep learning interpretability.

A. Model transparency

Much of the recent work on deep learning interpretability has focussed on understanding what the network has learned and why it has learned it; in other words, it addresses the dimensions of decomposability and algorithmic transparency. Given the size and complexity of deep models, the third dimension of transparency simulatability - is assumed to be very low. The references provided here thus relate only to the first two transparency dimensions.

Erhan et al. [6] developed one of the first methods for visualizing the responses of individual units in (unsupervised) deep belief networks. They developed methods for analyzing units in any layer of a network, while previous methods only looked at units in the first (input) layer. Zeiler and Fergus [7] extended this idea to (supervised) convolutional neural networks (CNNs) by using deconvolutional networks [8] – CNNs that map features to the input pixel space – to analyse higher-layer units. They used their visualizations to guide modifications to the CNN that improved its accuracy, and showed that having a minimum model depth was crucial to its performance. This work provides an important example of how increased transparency is not just important for understanding model behaviour; it can also guide us to build better models. Karpathy et al. [9] provided similar insights for recurrent neural networks (RNNs) – specifically Long-Short-Term-Memory (LSTM) RNNs. They trained an LSTM RNN one character at a time on different texts, and developed a method to show the activation of individual units as they generated new text. They showed that some cells learned easily-interpretable features in the text that spanned over a long time-range; for example, keeping track of quotations or line-lengths. Other units, though, produced less easily interpretable outputs, switching on and off with no easily discernible pattern.

Much further work on understanding higher-layer representations in deep models has focused on CNNs. Mahendran and Vedaldi [10] investigated the information contained in image representations at different CNN levels, revealing that deeper layers learn increasingly abstract representations of the image contents, thus making their responses more invariant to changes in the input image. Yosinski et al. [11] built on this work, improving the presentation of the image representations and releasing a software tool that provides several different visualizations designed to reveal the function that each neuron is performing within the network. Using similar methods, Nguyen et al. [12] showed that CNNs learn the global structure, details, and context of objects rather than a small number of local discriminating features.

Several groups have taken an alternative approach to understanding CNNs: generating the CNN's preferred image for each class it has learned. Simonyan et al. [13] provide an early example of this, generating images by maximizing the output score of the network for each class in turn. Their images qualitatively demonstrate the input features that most represent each class. Nguyen et al. [14] make use of a Deep Generator Network to generate preferred images for particular neurons in a CNN, producing very realistic synthetic images that they claim make their method more easily interpretable when trying to understand what a CNN has learned.

Another approach to understanding deep networks was developed by Li et al. [15], who focused on whether different networks learn similar features (convergent learning). Their method involves first training many networks, then analyzing the representations learned by each network at a per-neuron, or per-neuron-group level. They found that representations could be learned both by individual neurons and by groups of neurons, and that, while multiple networks reliably learn certain features, other features were distinct to individual networks. This work reveals that, while deep networks may show similar levels of performance, they can differ in what they learn from the training data.

Koh and Liang [16] propose a method to investigate a model from the point of view of its training data. They do this by asking how a model's predictions would differ if a

particular data point were altered, or not seen during training at all. They use a scaled-up derivation of statistical influence functions to approximate the effects of changing every training point without having to fully retrain the model. Their method provides a way of assessing the importance of particular training points on the classification of a test point, allowing the model-builder to find training points that contribute most to classification errors. This reveals how outliers can dominate learned model parameters, and potentially indicate mis-labeled training data. Additionally, they show it is possible to generate "adversarial" training images (images that are modified with noise such that the modification is imperceptible to a human, but results in a degradation in model performance). Prior to this work, adversarial examples had only been considered as inputs engineered to cause already-trained models to misclassify them [17], [18]; Koh and Liang show that classifiers can also be attacked through specially engineered training data.

A recent and promising approach due to Shwartz-Ziv and Tishby [19] provides a deeper insight into some of the above results by analyzing deep networks using information theory. They calculate how information is preserved on each layer's inputs and outputs using the Information Bottleneck framework [20]. Their method shows that the common stochastic gradient descent optimization method for learning parameters undergoes two separate phases during training. Early on (the "drift" phase), the variance of the weights' gradients is much smaller than the means of the gradients, indicating a high signal-to-noise ratio. Later during training (the "diffusion" phase), there is a rapid reversal such that the variance of the weights' gradients becomes greater than the means of the gradients, indicating a low signal-to-noise ratio. During this diffusion phase, fluctuations dominate the stochastic gradient descent and the error saturates. These results lead to a new interpretation of how stochastic gradient descent optimizes the network: compression by diffusion creates efficient internal representations. They also suggest that simpler stochastic diffusion algorithms could be used during the diffusion phase of training, reducing training time. Additionally, the results in [19] show that many different weight values can produce an optimally performing network, with implications for efforts to interpret single units. These results, along with explanations for the importance of network depth and the information bottleneck optimality of the layers, make Shwartz-Ziv and Tishby's work extremely promising for improving the transparency of deep learning, though their results so far are theoretical and their methods yet to be extended to real-world scenarios involving large networks and complex data.

B. Model Functionality

Model functionality can be explained by post-hoc interpretations of what the model has done. Lipton identifies four kinds of post-hoc explanation: textual (the model gives a justification for its output in text, or spoken language), visual (the model justifies its decision by some visualization method), local (the model justifies its decision in the context of the local feature space around the input) and by example (the model provides examples of similar inputs) [2]. All of these kinds of explanation have been explored by the deep learning community.

t-SNE (t-Distributed Stochastic Neighbourhood Embedding) [5] is a widely-used visualization method designed to show the data's inherent structure across multiple scales. Though not itself a "deep" method, t-SNE is frequently used alongside deep learning, as both deal well with high dimensional data. t-SNE visualizations can help in understanding the data and, therefore, what an algorithm might have learned, but does not directly explain the decisions of a particular algorithm.

Terrapattern [21] is a recent interactive tool for exploring visually similar areas in cities based on satellite images. The creators of Terrapattern trained a CNN using labelled satellite images. After this supervised learning phase, they removed the top classification layers of the network and used the remaining convolutional layers to generate features for each tile in their satellite images. They use these compressed feature representations to find the most similar-looking tiles to a particular query tile using k-Nearest Neighbours. The purpose of the system is to help users identify visually similar map locations rather than to make the CNN more interpretable, but by presenting examples of tiles that are close in the CNN's feature space, the user gains a post-hoc explanation of what the CNN has learned from the data.

Recent work by Hendricks et al. [22] attempts to provide a model that both classifies images, and provides accurate text explanations for why the image belongs to a particular category. Their description generation method is inspired by recent advances in automatic captioning techniques, which aim to provide appropriate text descriptions of images or videos. Hendricks et al. [22] expand on Donahue et al.'s captioning and recognition method [23]. Their model consists of a CNN that extracts visual features, combined with two LSTM RNNs that learn to create a caption. The first RNN, trained on the image descriptions, generates words based only on the previously generated word, while the second RNN is fed the first RNN's output, the image features, and the image category predicted by the CNN. The second RNN then generates the next word conditioned on this input. They show that this method generates image and class-relevant explanations for classification decisions on a difficult bird-species discrimination task. The results are impressive; however, the model does not guarantee that the descriptions it learns will correspond to the visual features that humans interpret them as referring to, and does not provide a way to check this. By contrast, Xu et al.'s caption generation method [24] can show where in the image the network is focusing its attention while generating each word in its description, but does not perform classification.

Several groups have developed methods for identifying and visualizing the features in individual test data points that contribute the most towards a classifier's output (i.e. local explanation). Perhaps the most well-known method is Ribeiro et al.'s Local Interpretable Model-Agnostic Explanations (LIME), an algorithm that provides explanations of decisions for any

machine learning model [25]. The LIME algorithm outputs a binary vector representing the input: each bit corresponds to an input feature (e.g. a word in a document, or a contiguous region – super-pixel – in an image), with ones indicating that the feature was important for the classifier's output, and a zero indicating it was unimportant. It calculates the importance of each feature by generating perturbed samples of the input point and using these samples (labeled by the original model) to learn a local approximation to the model. LIME can be particularly helpful in identifying confusing input features, allowing for dataset debugging or improved feature engineering. LIME works with any model, and was tested on a CNN in [25]. However, its sampling approach means it can be too slow for interactive use with complex models. Recent work by Elenberg et al. develops an alternative method producing similar outputs using an efficient streaming algorithm, improving the speed of explanation generation up to 10x over LIME [26].

Another recent method, due to Ross et al. [27] and inspired by LIME, allows a developer to constrain their model during training to be "right for the right reasons" (RRR) rather than learning spurious correlations in the training data. RRR works using binary masks that specify whether an input feature should be irrelevant to the classification of that example, as assessed by a human expert. The authors additionally propose an automatic method that learns a set of models using different masks designed to create models with different decision boundaries. This automated approach provides a way of quantifying ambiguity in the training data by counting the number of different models that can be learned without reducing accuracy.

Turning to methods developed specifically for interpreting deep models, the recently-proposed layer-wise relevance propagation (LRP) algorithm from Wojciech Samek's group [28], [29] uses the fact that the individual neural network units are differentiable to decompose the network output in terms of its input variables. It is a principled method that has a close relationship to Taylor decomposition and is applicable to arbitrary deep neural network architectures [30]. The output is a heatmap over the input features that indicates the relevance of each feature to the model output. This makes the method particularly well suited to analyzing image classifiers, though the method has also been adapted for text and electroencephalogram signal classification [31]. Samek et al. [32] have also developed an objective metric for comparing the output of LRP with similar heatmapping algorithms. Kumar et al. [33] present an alternative heat-mapping method that can show the image regions that the model was most attentive to, but also allows for multiple classes to be associated with these regions of attention, whereas LRP assumes all features make either a zero or positive contribution to the single predicted class.

Finally, Lei et al. [34] developed a local explanation approach that reveals the most relevant sentences in sentiment prediction from text documents. Their method combines two modular components – a generator and encoder – that operate together and learn candidate rationales for a prediction. Rationales are simply subsets of the words from the input text that satisfy two properties: the selected words represent short, coherent pieces of text (e.g., phrases), and the selected words alone must result in the same prediction as the whole original text. For a given input text, the generator specifies a distribution over possible rationales. The encoder then maps the rationale to task specific values. The distribution that minimizes the regularized encoder loss function is used as the rationale.

III. A coalition perspective

We consider the problem of model interpretation within a coalition setting in which multiple disparate parties come together to forge an ad-hoc coalition geared towards achieving a common mission. Each party owns a slice of data but has policy-based constraints that places restrictions on the information that it can share with other coalition members. The success of the mission is thus contingent upon maximum utilization of this distributed data to build a common model shared among all the parties.

As is evident from the above setting, any decision made using the common model has to be adequately justified for it to be accepted by all the coalition members. Such a justification can only be generated using an interpretable model. In addition, it is quintessential that the common model is established as fair (i.e., unbiased), accountable and transparent to the coalition members. Finally, the policy-constraints within a coalition together with the non-homogeneity between the model architectures might make it difficult to use techniques such as layer-wise relevance propagation for interpretation.

IV. Discussion and Challenges

We now discuss in detail challenges unique to coalition and possible alternative approaches to providing interpretability.

A. Fairness and Accountability

With rapid adoption of machine learning techniques there has also been a growing recognition that the same techniques also raise novel ethical, policy, and legal challenges. In particular, policymakers, regulators, and advocates have expressed fears about the potentially discriminatory impact of data-driven learning approaches, especially the dangers of inadvertently encoding bias into automated decisions. At the same time, there is an increasing alarm that the complexity of machine learning and opaqueness of data mining processes may reduce the justification for consequential decisions to "the algorithm made me do it" or "this is what the model says."

To begin, a formal definition of *fairness* is in order when fairness becomes a machine learning objective. What does it mean for an algorithm to be fair, possibly in the presence of (e.g.: human, social, contextual etc.) bias in the the dataset? [35]. Motivated by these concerns, Matthew et. al. study a technical definition of fairness modeled after Rawl's notion of "fair equality of opportunity" [36]. They also introduce the notion of a "discrimination index", and show that

standard algorithms for our problem exhibit structured discriminatory behavior. Yang et. al. [37] consider the problem of ranking a set of individuals bvased on demographic, behavioral or other characteristics wherein rankers can, and often do, discriminate against individuals and disadvantaged members of protected groups despite seemingly automatic and objective metrics [38]. The authors formulate a fairness measure by taking several well-known statistical parity measures proposed in literature and make them rank-aware by placing them within well-known IR evaluation techniques [39].

We would need to develop new computational techniques for discrimination-aware data mining [40]. Legal definitions and distributional constraints and geographic parity can be good starting points but how to translate them to practical algorithmic contexts remains an open question. Needless to mention, turning fairness into a computational problem may bring us back to the same place we started at, so we need to think about how we can keep fairness measures, *fair*.

Accountability may be viewed as the ability to inspect a model in post hoc, and make it available for human or algorithmic inspection. Many important decisions histiorically made by humans are now being made by algorithms, whose accountability measures and legal standards are far from satisfactory [41]. While model transparency is important, it is neither necessary nor sufficient in the industry. Source code is often propreitary, and transparency may be undesirable where private information or security are of concern. Accountability is arguably more important, even in the absence of model transparency. Adler et. al. present a technique for auditing black-box models, which can reveal the extent to which the models take advantage of particular features in the dataset [42], without knowing how the models work [43]. While there may be technical challenges [44] in allowing public auditing while protecting proprietary information, private auditing may be the right option, for which methods need to be developed [43], [45], [46]. Practical methods to test machine learning algorithms for policy compliance need to be developed, with the expectation that we can prove that an algorithm behaves in a certain way without necessarily having to reveal the algorithm. Another desirable behavior in this respect is for the model to be able to state its goals, and for someone other than the creator to be able to verify that these goals are achieved and if not, be able to demonstrate the causal origins of the outcome predicted by the model.

B. Interpretability versus Explainability

Computational models that impart reasoning behind their decisions, often use the terms "interpretability" and "explainability" synonymously [16], [47], [48]. This is true, even when the community acknowledges the need for clear taxonomy [2]. We would like to propose a differentiation between these terms and in doing so, we are able to clarify the process of forming testable metrics within the problem space.

When talking about the explainability of a model, we suggest that this refers specifically to the type and completeness of the output given when a model is queried for reasoning behind its decision. This means that explanations of the same type can be compared using a metric without need for any further context [32]. However, explanations of different types (saliency map images [13] and text captions for example [22]) can't be compared using a metric.

Interpretability, we would suggest, specifically refers to the interpretation arrived at by a user agent when given an explanation from a model. Thus, applying a metric to the interpretability of a model must be done with a context, e.g. the task the model is being used for, the knowledge and experience of the user agent, the specific query that requires an explanation etc. By adding this context, explanations of different types (more specifically the interpretations they lead to) can be compared with a metric.

We illustrate this difference with the following use case. Given a scenario in which a model is predicting accurately but the decision maker also demands a high level of confidence in the intelligence being provided by the system, an explanation from the model must lead to an interpretation that gives strong reasoning (in the eyes of the user agent) to the conclusion of the model. However, in a scenario where a skilled agent is looking to debug an erroneous classification, the explanation must point more closely to the underlying architecture of the model to allow for an interpretation to form of what might need changing in the model to improve it. Further motivation for the differentiation between the terms can be seen when the user agent from the first use case is placed in to the second use case. The explanation offered would be unchanged in terms of its type and quality but the interpretation is likely to be of a lower quality because the user agent now lacks the knowledge to utilize the explanation.

C. Bayesian approach to interpretability

Compared to deep learning approaches, Bayesian reasoning provides a unified framework for model building, inference, prediction and decision making. There is explicit accounting for uncertainty and variability of outcomes. Finally, the framework is also robust to model overfitting and Bayes rule provides an automatic "Occam's Razor" effect, penalizing unnecessarily complex models. However, for reasons of computational tractability of inferences, Bayesian reasoning is restricted primarily to conjugate and linear models.

The above leads us to the observation that there exists elements in the Bayesian reasoning and deep learning frameworks that complement each other. This observation has been exploited in recent work on Bayesian Deep Learning (BDL) in particular [49] aiming to integrate deep learning and Bayesian models within a uniform probabilistic framework. Such a neural network can help interpretability in terms of both model transparency and model functionality.

ACKNOWLEDGEMENTS

This research was sponsored by the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted

as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the UK Ministry of Defence or the UK Government. The U.S. and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation hereon.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] Z. C. Lipton, "The mythos of model interpretability," *CoRR*, vol. abs/1606.03490, 2016.

[3] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.

[4] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *CoRR*, vol. abs/1604.00289, 2016.

[5] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[6] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing Higher-Layer Features of a Deep Network," University of Montreal, Tech. Rep. 1341, Jun. 2009.

[7] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision ECCV 2014*. Springer, Cham, Sep. 2014, pp. 818–833.

[8] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2018–2025.

[9] A. Karpathy, J. Johnson, and F. Li, "Visualizing and understanding recurrent networks," *CoRR*, vol. abs/1506.02078, 2015.

[10] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 5188–5196.

[11] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding Neural Networks Through Deep Visualization," *arXiv:1506.06579 [cs]*, Jun. 2015, arXiv: 1506.06579.

[12] A. Nguyen, J. Yosinski, and J. Clune, "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks," *arXiv:1602.03616 [cs]*, Feb. 2016.

[13] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv:1312.6034 [cs]*, Dec. 2013.

[14] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 3387–3395.

[15] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft, "Convergent Learning: Do different neural networks learn the same representations?" in *PMLR*, Dec. 2015, pp. 196–212.

[16] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," *arXiv preprint arXiv:1703.04730*, 2017.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *arXiv preprint arXiv:1610.08401*, 2016.

[19] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *arXiv preprint arXiv:1703.00810*, 2017.

[20] N. Tishby, F. C. Pereira, and W. Bialek, "The Information Bottleneck Method," *Proceedings of 37th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

[21] G. Levin, D. Newbury, and K. McDonald, "Terrapattern." [Online]. Available: http://www.terrapattern.com

[22] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating visual explanations," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–19.

[23] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.

[26] E. R. Elenberg, A. G. Dimakis, M. Feldman, and A. Karbasi, "Streaming weak submodularity: Interpreting neural networks on the fly," *arXiv preprint arXiv:1703.02647*, 2017.

[27] A. S. Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the right reasons: Training differentiable models by constraining their explanations," *arXiv preprint arXiv:1703.03717*, 2017.

[28] A. Binder, G. Montavon, S. Bach, K. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," *CoRR*, vol. abs/1604.00825, 2016.

[29] A. Binder, S. Bach, G. Montavon, K.-R. Müller, and W. Samek, *Layer-Wise Relevance Propagation for Deep Neural Network Architectures*. Springer, 2016.

[30] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[31] I. Sturm, S. Bach, W. Samek, and K. Müller, "Interpretable deep neural networks for single-trial EEG classification," *CoRR*, vol. abs/1604.08201, 2016.

[32] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE transactions on neural networks and learning systems*, 2017.

[33] D. Kumar, A. Wong, and G. W. Taylor, "Explaining the unexplained: A class-enhanced attentive response (clear) approach to understanding deep neural networks," *arXiv preprint arXiv:1704.04133*, 2017.

[34] T. Lei, R. Barzilay, and T. S. Jaakkola, "Rationalizing neural predictions," *CoRR*, vol. abs/1606.04155, 2016.

[35] K. Lum and W. Isaac, "To predict and serve?" *Significance*, vol. 13, no. 5, pp. 14–19, 2016.

[36] M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, "Rawlsian fairness for machine learning," *CoRR*, vol. abs/1610.09559, 2016.

[37] K. Yang and J. Stoyanovich, "Measuring fairness in ranked outputs," *CoRR*, vol. abs/1610.08559, 2016.

[38] I. Zliobaite, "A survey on measuring indirect discrimination in machine learning," *CoRR*, vol. abs/1511.00148, 2015.

[39] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002.

[40] S. Hajian, J. Domingo-Ferrer, A. Monreale, D. Pedreschi, and F. Giannotti, "Discrimination- and privacy-aware patterns," *Data Min. Knowl. Discov.*, vol. 29, no. 6, Nov. 2015.

[41] J. A. Kroll, J. Huey, S. Barocas, E. W. Felten, J. R. Reidenberg, D. G. Robinson, and H. Yu, "Accountable algorithms," 2017.

[42] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[43] P. Adler, C. Falk, S. A. Friedler, G. Rybeck, C. Scheidegger, B. Smith, and S. Venkatasubramanian, "Auditing black-box models for indirect influence," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec 2016, pp. 1–10.

[44] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," *CoRR*, vol. abs/1609.02943, 2016.

[45] W. Duivesteijn and J. Thaele, "Understanding where your classifier does (not) work – the scape model class for emm," in *2014 IEEE International Conference on Data Mining*, Dec 2014, pp. 809–814.

[46] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics*, vol. 8, no. 1, 2007.

[47] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911*, 2017.

[48] S. Jha, V. Raman, A. Pinto, T. Sahai, and M. Francis, "On learning sparse boolean formulae for explaining ai decisions," 2017.

[49] H. Wang and D.-Y. Yeung, "Towards bayesian deep learning: A survey," *arXiv preprint arXiv:1604.01662*, 2016.