Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers

Fred Hohman, *Member, IEEE*, Minsuk Kahng, *Member, IEEE*,
Robert Pienta, *Member, IEEE*, and Duen Horng Chau, *Member, IEEE*

Abstract—Deep learning has recently seen rapid development and received significant attention due to its state-of-the-art performance on previously-thought hard problems. However, because of the internal complexity and nonlinear structure of deep neural networks, the underlying decision making processes for why these models are achieving such performance are challenging and sometimes mystifying to interpret. As deep learning spreads across domains, it is of paramount importance that we equip users of deep learning with tools for understanding when a model works correctly, when it fails, and ultimately how to improve its performance. Standardized toolkits for building neural networks have helped democratize deep learning; visual analytics systems have now been developed to support model explanation, interpretation, debugging, and improvement. We present a survey of the role of visual analytics in deep learning research, which highlights its short yet impactful history and thoroughly summarizes the state-of-the-art using a human-centered interrogative framework, focusing on the *Five W's and How* (Why, Who, What, How, When, and Where). We conclude by highlighting research directions and open research problems. This survey helps researchers and practitioners in both visual analytics and deep learning to quickly learn key aspects of this young and rapidly growing body of research, whose impact spans a diverse range of domains.

Index Terms—Deep learning, visual analytics, information visualization, neural networks

✦

1 INTRODUCTION

DEEP learning is a specific set of techniques from the broader field of machine learning (ML) that focus on the study and usage of *deep* artificial neural networks to learn structured representations of data. First mentioned as early as the 1940s [1], artificial neural networks have a rich history [2], and have recently seen a dominate and pervasive resurgence [3], [4], [5] in many research domains by producing state-of-the-art results [6], [7] on a number of diverse big data tasks [8], [9]. For example, the premiere machine learning, deep learning, and artificial intelligence (AI) conferences have seen enormous growth in attendance and paper submissions since early 2010s. Furthermore, open-source toolkits and programming libraries for building, training, and evaluating deep neural networks have become more robust and easy to use, democratizing deep learning. As a result, the barrier to developing deep learning models is lower than ever before and deep learning applications are becoming pervasive.

While this technological progress is impressive, it comes with unique and novel challenges. For example, the lack of interpretability and transparency of neural

networks, from the learned representations to the underlying decision process, is an important problem to address. Making sense of why a particular model misclassifies test data instances or behaves poorly at times is a challenging task for model developers. Similarly, end-users interacting with an application that relies on deep learning to make critical decisions may question its reliability if no explanation is given by the model, or become baffled if the explanation is convoluted. While explaining neural network decisions is important, there are numerous other problems that arise from deep learning, such as AI safety and security (e.g., when using models in applications such as self-driving vehicles), and compromised trust due to bias in models and datasets, just to name a few. These challenges are often compounded, due to the large datasets required to train most deep learning models. As worrisome as these problems are, they will likely become even more widespread as more AI-powered systems are deployed in the world. Therefore, a general sense of model understanding is not only beneficial, but often required to address the aforementioned issues.

Data visualization and visual analytics excel at knowledge communication and insight discovery by using encodings to transform abstract data into meaningful representations. In the seminal work by Zeiler and Fergus [10], a technique called *deconvolutional networks* enabled projection from a model's learned feature space back to the pixel space. Their technique and results give insight into what types of features deep neural networks are learning at specific layers, and also serve as a debugging tool for improving a model. This work is often credited for popularizing visualization in the machine

• The authors are with the College of Computing, Georgia Tech, Atlanta, Georgia 30332.
  E-mail: {fredhohman, kahng, pientars, polo}@gatech.edu.

Visual Analytics in Deep Learning │ Interrogative Survey Overview

§4 WHY

Why would one want to use visualization in deep learning?

Interpretability & Explainability
Debugging & Improving Models
Comparing & Selecting Models
Teaching Deep Learning Concepts

§6 WHAT

What data, features, and relationships in deep learning can be visualized?

Computational Graph & Network Architecture
Learned Model Parameters
Individual Computational Units
Neurons In High-dimensional Space
Aggregated Information

§8 WHEN

When in the deep learning process is visualization used?

During Training
After Training

§5 WHO

Who would use and benefit from visualizing deep learning?

Model Developers & Builders
Model Users
Non-experts

§7 HOW

How can we visualize deep learning data, features, and relationships?

Node-link Diagrams for Network Architecture
Dimensionality Reduction & Scatter Plots
Line Charts for Temporal Metrics
Instance-based Analysis & Exploration
Interactive Experimentation
Algorithms for Attribution & Feature Visualization

§9 WHERE

Where has deep learning visualization been used?

Application Domains & Models
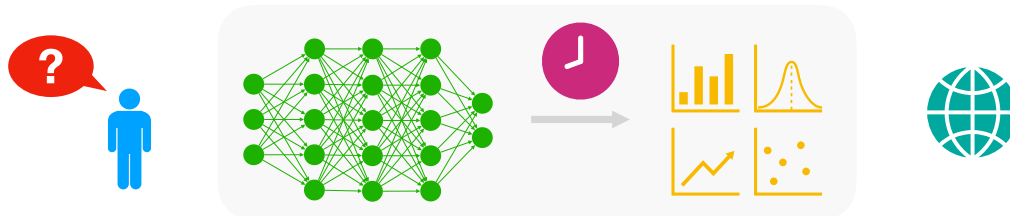A Vibrant Research Community

Fig. 1. A visual overview of our interrogative survey, and how each of the six questions, "Why, Who, What, How, When, and Where," relate to one another. Each question corresponds to one section of this survey, indicated by the numbered tag, near each question title. Each section lists its major sections discussed in the survey.

learning and computer vision communities in recent years, putting a spotlight on it as a powerful tool that helps people understand and improve deep learning models. However, visualization research for neural networks started well before [11], [12], [13]. Over just a handful of years, many different techniques have been introduced to help interpret what neural networks are learning. Many such techniques generate static images, such as attention maps and heatmaps for image classification, indicating which parts of an image are most important to the classification. However, interaction has also been incorporated into the model understanding process in visual analytics tools to help people gain insight [14], [15], [16]. This hybrid research area has grown in both academia and industry, forming the basis for many new research papers, academic workshops, and deployed industry tools.

In this survey, we summarize a large number of deep learning visualization works using the *Five W's and How* (Why, Who, What, How, When, and Where). Fig. 1 presents a visual overview of how these interrogative questions reveal and organize the various facets of deep learning visualization research and their related topics. By framing the survey in this way, many existing works fit a description as the following fictional example:

> *To interpret representations learned by deep models (why), model developers (who) visualize neuron activations in convolutional neural networks (what) using t-SNE embeddings (how) after the training phase (when) to solve an urban planning problem (where).*

This framing captures the needs, audience, and techniques of deep learning visualization, and positions new work's contributions in the context of existing literature.

We conclude by highlighting prominent research directions and open problems. We hope that this survey acts as a companion text for researchers and practitioners wishing to understand how visualization supports deep learning research and applications.

2 OUR CONTRIBUTIONS & METHOD OF SURVEY

2.1 Our Contributions

C1. We present a comprehensive, timely survey on visualization and visual analytics in deep learning research, using a human-centered, interrogative framework. This method enables us to position each work with respect to its *Five Ws and How* (Why, Who, What, How, When, and Where), and flexibly discuss and highlight existing works' multifaceted contributions.

- Our human-centered approach using the *Five W's and How* — based on how we familiarize ourselves with new topics in everyday settings—enables readers to quickly grasp important facets of this young and rapidly growing body of research.
- Our interrogative process provides a framework to describe existing works, as well as a model to base new work off of.

C2. To highlight and align the cross-cutting impact that visual analytics has had on deep learning across a

broad range of domains, our survey goes beyond visualization-focused venues, extending a wide scope that encompasses most relevant works from many top venues in artificial intelligence, machine learning, deep learning, and computer vision. We highlight how visual analytics has been an integral component in solving some of AI's biggest modern problems, such as neural network interpretability, trust, and security.

C3. As deep learning, and more generally AI, touches more aspects of our daily lives, we highlight important research directions and open problems that we distilled from the survey. These include improving the capabilities of visual analytics systems for furthering interpretability, conducting more effective design studies for evaluating system usability and utility, advocating humans' important roles in AI-powered systems, and promoting proper and ethical use of AI applications to benefit society.

2.2 Survey Methodology & Summarization Process

We selected existing works from top computer science journals and conferences in visualization (e.g., IEEE Transactions on Visualization and Computer Graphics (TVCG)), visual analytics (e.g., IEEE Conference on Visual Analytics Science and Technology (VAST)) and deep learning (e.g., Conference on Neural Information Processing Systems (NIPS) and the International Conference on Machine Learning (ICML)). Since deep learning visualization is relatively new, much of the relevant work has appeared in workshops at the previously mentioned venues; therefore, we also include those works in our survey. Table 1 lists some of the most prominent publication venues and their acronyms. We also inspected preprints posted on arXiv (https://arxiv.org/), an open-access, electronic repository of manuscript preprints, whose computer science subject has become a hub for new deep learning research. Finally, aside from the traditional aforementioned venues, we include non-academic venues with significant attention such as Distill, industry lab research blogs, and research blogs of influential figures. Because of the rapid growth of deep learning research and the lack of a perfect fit for publishing and disseminating work in this hybrid area, the inclusion of these non-traditional sources are important to review, as they are highly influential and impactful to the field.

Visualization takes many forms throughout the deep learning literature. This survey focuses on visual analytics for deep learning. We also include related works from the AI and computer vision communities that contribute novel static visualizations. So far, the majority of work surrounds convolutional neural networks (CNNs) and image data; more recent work has begun to visualize other models, e.g., recurrent neural networks (RNNs), long short-term memory units (LSTMs), and generative adversarial networks (GANs). For each work, we recorded the following information if present:

- Metadata (title, authors, venue, and year published)
- General approach and short summary
- Explicit contributions
- Future work

TABLE 1
Relevant Visualization and AI Venues, in the Order of:
Journals, Conferences, Workshops, Open Access Journals,
and Preprint Repositories. In Each Category,
Visualization Venues Precede AI Venues.

TVCG	IEEE Transactions on Visualization and Computer Graphics
VAST	IEEE Conference on Visual Analytics Science and Technology
InfoVis	IEEE Information Visualization
VIS	IEEE Visualization Conference (VAST+InfoVis+SciVis)
CHI	ACM Conference on Human Factors in Computing Systems
NIPS	Conference on Neural Information Processing Systems
ICML	International Conference on Machine Learning
CVPR	Conference on Computer Vision and Pattern Recognition
ICLR	International Conference on Learning Representations
VADL	IEEE VIS Workshop on Visual Analytics for Deep Learning
HCML	CHI Workshop on Human Centered Machine Learning
IDEA	KDD Workshop on Interactive Data Exploration & Analytics
	ICML Workshop on Visualization for Deep Learning
WHI	ICML Workshop on Human Interpretability in ML
	NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning
	NIPS Interpretable ML Symposium
FILM	NIPS Workshop on Future of Interactive Learning Machines
	ACCV Workshop on Interpretation and Visualization of Deep Neural Nets
	ICANN Workshop on Machine Learning and Interpretability
Distill	Distill: Journal for Supporting Clarity in Machine Learning
arXiv	arXiv.org e-Print Archive

- Design component (e.g., user-centered design methodologies, interviews, evaluation)
- Industry involvement and open-source code

With this information, we used the *Five W's and How* (Why, Who, What, How, When, and Where) to organize these existing works and the current state-of-the-art of visualization and visual analytics in deep learning.

2.3 Related Surveys

While there is a larger literature for visualization for machine learning, including predictive visual analytics [17], [18], [19] and human-in-the-loop interactive machine learning [20], [21], to our knowledge there is no comprehensive survey of visualization and visual analytics for deep learning. Regarding deep neural networks, related surveys include a recent book chapter that discusses visualization of deep neural networks related to the field of computer vision [22], an unpublished essay that proposes a preliminary taxonomy for visualization techniques [23], and an article that focuses on describing interactive model analysis, which mentions deep learning in a few contexts while describing a high-level framework for general machine learning models [24]. A recent overview article by Choo and Liu [25] is the closest in spirit to our survey. Our survey provides wider coverage and more detailed analysis of the literature.

Different from all the related articles mentioned above, our survey provides a comprehensive, human-centered, and interrogative framework to describe deep learning visual analytics tools, discusses the new, rapidly growing

community at large, and presents the major research trajectories synthesized from existing literature.

2.4 Survey Overview & Organization

Section 3 introduces common deep learning terminology. Fig. 1 shows a visual overview of this survey's structure and Table 2 summarizes representative works. Each interrogative question (Why, Who, What, How, When, and Where) is given its own section for discussion, ordered to best motivate why visualization and visual analytics in deep learning is such a rich and exciting area of research.

§ 4 Why do we want to visualize deep learning?
Why and for what purpose would one want to use visualization in deep learning?

§ 5 Who wants to visualize deep learning?
Who are the types of people and users that would use and stand to benefit from visualizing deep learning?

§ 6 What can we visualize in deep learning?
What data, features, and relationships are inherent to deep learning that can be visualized?

§ 7 How can we visualize deep learning?
How can we visualize the aforementioned data, features, and relationships?

§ 8 When can we visualize deep learning?
When in the deep learning process is visualization used and best suited?

§ 9 Where is deep learning visualization being used?
Where has deep learning visualization been used?

Section 10 presents research directions and open problems that we gathered and distilled from the literature survey. Section 11 concludes the survey.

3 COMMON TERMINOLOGY

To enhance readability of this survey, and to provide quick references for readers new to deep learning, we have tabulated a sample of relevant and common deep learning terminology used in this work, shown in Table 3. The reader may want to refer to Table 3 throughout this survey for technical terms, meanings, and synonyms used in various contexts of discussion. The table serves as an introduction and summarization of the state-of-the-art. For definitive technical and mathematical descriptions, we encourage the reader to refer to excellent texts on deep learning and neural network design, such as the *Deep Learning* textbook [26].

4 WHY VISUALIZE DEEP LEARNING

4.1 Interpretability & Explainability

The most abundant, and to some, the most important reason why people want to visualize deep learning is to understand how deep learning models make decisions and what representations they have learned, so we can place trust in a model [60]. This notion of general model understanding has been called the *interpretability* or *explainability* when referring to machine learning models [60], [61], [62]. However, neural networks particularly suffer from this problem since oftentimes real world and high-performance models contain a large number of parameters (in the millions) and exhibit extreme internal complexity by using many non-linear transformations at different stages during training. Many works

motivate this problem by using phrases such as "opening and peering through the black-box," "transparency," and "interpretable neural networks," [13], [56], [63], referring the internal complexity of neural networks.

4.1.1 Discordant Definitions for Interpretability

Unfortunately, there is no universally formalized and agreed upon definition for explainability and interpretability in deep learning, which makes classifying and qualifying interpretations and explanations troublesome. In Lipton's work "The Mythos of Model Interpretability [60]," he surveys interpretability-related literature, and discovers diverse motivations for why interpretability is important and is occasionally discordant. Despite this ambiguity, he attempts to refine the notion of interpretability by making a first step towards providing a comprehensive taxonomy of both the desiderata and methods in interpretability research. One important point that Lipton makes is the difference between interpretability and an explanation; an explanation can show predictions without elucidating the mechanisms by which models work [60].

In another work originally presented as a tutorial at the International Conference on Acoustics, Speech, and Signal Processing by Montavona et al. [61], the authors propose exact definitions of both an interpretation and an explanation. First, an interpretation is "the mapping of an abstract concept (e.g., a predicted class) into a domain that the human can make sense of." They then provide some examples of interpretable domains, such as images (arrays of pixels) and text (sequences of words), and noninterpretable domains, such as abstract vector spaces (word embeddings). Second, an explanation is "the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g., classification or regression)." For example, an explanation can be a heatmap highlighting which pixels of the input image most strongly support an image classification decision, or in natural language processing, explanations can highlight certain phrases of text.

However, both of the previous works are written by members of the AI community, whereas work by Miller titled "Explanation in Artificial Intelligence: Insights from the Social Sciences" [62] postulates that much of the current research uses only AI researchers' intuition of what constitutes a "good" explanation. He suggests that if the focus on explaining decisions or actions to a human observer is the goal, then if these techniques are to succeed, the explanations they generate should have a structure that humans accept. Much of Miller's work highlights vast and valuable bodies of research in philosophy, psychology, and cognitive science for how people define, generate, select, evaluate, and present explanations, and he argues that interpretability and explainability research should leverage and build upon this history [62]. In another essay, Offert [64] argues that to make interpretability more rigorous, we must first identify where it is impaired by intuitive considerations. That is, we have to "consider it precisely in terms of what it is not." While multiple works bring different perspectives, Lipton makes the keen observation that for the field to progress, the community must critically engage with this problem formulation issue [60]. Further research will help solidify the notions of interpretation and explanation.

TABLE 2
Overview of Representative Works in Visual Analytics for Deep Learning. Each Row is One Work; Works are Sorted Alphabetically by First Author's Last Name. Each Column Corresponds to A Section from The Six Interrogative Questions. A Work's Relevant Section is Indicated by A Colored Cell.

Column legend:

WHY — 4.1 Interpretability & Explainability; 4.2 Debugging & Improving Models; 4.3 Comparing & Selecting Models; 4.4 Teaching Deep Learning Concepts
WHO — 5.1 Model Developers & Builders; 5.2 Model Users; 5.3 Non-experts
WHAT — 6.1 Computational Graph & Network Architecture; 6.2 Learned Model Parameters; 6.3 Individual Computational Units; 6.4 Neurons in High-dimensional Space; 6.5 Aggregated Information
HOW — 7.1 Node-link Diagrams for Network Architecture; 7.2 Dimensionality Reduction & Scatter Plots; 7.3 Line Charts for Temporal Metrics; 7.4 Instance-based Analysis & Exploration; 7.5 Interactive Experimentation; 7.6 Algorithms for Attribution & Feature Visualization
WHEN — 8.1 During Training; 8.2 After Training
WHERE — 9.2 Publication Venue

Work	4.1	4.2	4.3	4.4	5.1	5.2	5.3	6.1	6.2	6.3	6.4	6.5	7.1	7.2	7.3	7.4	7.5	7.6	8.1	8.2	9.2
Abadi, et al., 2016 [27]	●	●	●		●	●						●			●				●	●	arXiv
Bau, et al., 2017 [28]	●		●		●					●		●				●		●		●	CVPR
Bilal, et al., 2017 [29]	●	●			●					●		●				●		●	●		TVCG
Bojarski, et al., 2016 [30]	●	●			●					●		●				●		●	●	●	arXiv
Bruckner, 2014 [31]	●	●			●			●	●				●			●		●	●		MS Thesis
Carter, et al., 2016 [32]	●			●	●	●	●			●	●	●				●	●			●	Distill
Cashman, et al., 2017 [33]	●	●			●	●				●	●					●				●	VADL
Chae, et al., 2017 [34]	●	●			●					●		●			●	●			●		VADL
Chung, et al., 2016 [35]	●	●			●			●		●	●		●	●	●	●			●		FILM
Goyal, et al., 2016 [36]	●						●		●							●	●	●		●	arXiv
Harley, 2015 [37]	●		●				●	●					●			●		●		●	ISVC
Hohman, et al., 2017 [38]	●		●	●			●					●				●		●	●		CHI
Kahng, et al., 2018 [39]	●	●			●	●		●				●	●	●		●				●	TVCG
Karpathy, et al., 2015 [40]	●				●	●				●	●	●				●				●	arXiv
Li, et al., 2015 [41]	●				●	●				●	●	●				●				●	arXiv
Liu, et al., 2017 [14]	●	●			●			●	●	●		●	●			●				●	TVCG
Liu, et al., 2018 [42]	●	●			●			●	●	●		●				●	●		●		TVCG
Ming, et al., 2017 [43]	●		●		●					●		●				●				●	VAST
Norton & Qi, 2017 [44]	●	●		●	●	●	●									●	●			●	VizSec
Olah, 2014 [45]	●		●				●			●				●		●	●			●	Web
Olah, et al., 2018 [46]	●		●		●	●	●	●		●						●	●	●		●	Distill
Pezzotti, et al., 2017 [47]	●	●			●					●	●	●		●	●	●			●		TVCG
Rauber, et al., 2017 [48]	●	●	●		●					●	●	●		●		●			●	●	TVCG
Robinson, et al., 2017 [49]	●				●	●				●	●	●				●				●	GeoHum.
Rong & Adar, 2016 [50]	●	●			●	●				●						●				●	ICML VIS
Smilkov, et al., 2016 [51]						●				●	●	●		●		●				●	NIPS WS.
Smilkov, et al., 2017 [16]	●	●		●			●	●	●			●	●		●		●		●	●	ICML VIS
Strobelt, et al., 2018 [52]	●	●			●	●				●		●			●	●				●	TVCG
Tzeng & Ma, 2005 [13]	●				●			●	●			●	●		●					●	VIS
Wang, et al., 2018 [53]	●	●	●		●					●	●		●			●			●		TVCG
Webster, et al., 2017 [54]			●			●										●	●		●	●	Web
Wongsuphasawat, et al., 2018 [15]		●			●			●				●	●								TVCG
Yosinski, et al., 2015 [55]	●		●		●	●		●		●						●	●	●		●	ICML DL
Zahavy, et al., 2016 [56]	●				●					●	●	●				●				●	ICML
Zeiler, et al., 2014 [10]	●	●			●					●	●							●		●	ECCV
Zeng, et al., 2017 [57]	●		●		●			●		●						●			●		VADL
Zhong, et al., 2017 [58]	●	●			●					●	●	●		●	●	●		●	●		ICML VIS
Zhu, et al., 2016 [59]	●				●	●	●					●				●	●	●		●	ECCV

TABLE 3
Foundational Deep Learning Terminology used in this Paper, Sorted by Importance.
In a Term's "Meaning" (Last Column), Defined Terms are Italicized.

Technical Term	Synonyms	Meaning
Neural Network	Artificial neural net, net	Biologically-inspired models that form the basis of deep learning; approximate functions dependent upon a large and unknown amount of inputs consisting of *layers* of *neurons*
Neuron	Computational unit, node	Building blocks of *neural networks*, entities that can apply *activation functions*
Weights	Edges	The trained and updated parameters in the *neural network* model that connect *neurons* to one another
Layer	Hidden layer	Stacked collection of *neurons* that attempt to extract features from data; a *layer's* input is connected to a previous *layer's* output
Computational Graph	Dataflow graph	Directed graph where nodes represent operations and edges represent data paths; when implementing *neural network* models, often times they are represented as these
Activation Functions	Transform function	Functions embedded into each *layer* of a *neural network* that enable the network represent complex non-linear decisions boundaries
Activations	Internal representation	Given a trained network one can pass in data and recover the *activations* at any *layer* of the network to obtain its current representation inside the network
Convolutional Neural Network	CNN, convnet	Type of *neural network* composed of convolutional *layers* that typically assume image data as input; these *layers* have depth unlike typical *layers* that only have width (number of *neurons* in a *layer*); they make use of filters (feature & pattern detectors) to extract spatially invariant representations
Long Short-Term Memory	LSTM	Type of *neural network*, often used in text analysis, that addresses the vanishing gradient problem by using memory gates to propagate gradients through the network to learn long-range dependencies
Loss Function	Objective function, cost function, error	Also seen in general ML contexts, defines what success looks like when learning a representation, i.e., a measure of difference between a *neural network's* prediction and ground truth
Embedding	Encoding	Representation of input data (e.g., images, text, audio, time series) as vectors of numbers in a high-dimensional space; oftentimes reduced so data points (i.e., their vectors) can be more easily analyzed (e.g., compute similarity)
Recurrent Neural Network	RNN	Type of *neural network* where recurrent connections allow the persistence (or "memory") of previous inputs in the network's internal state which are used to influence the network output
Generative Adversarial Networks	GAN	Method to conduct unsupervised learning by pitting a generative network against a discriminative network; the first network mimics the probability distribution of a training dataset in order to fool the discriminative network into judging that the generated data instance belongs to the training set
Epoch	Data pass	A complete pass through a given dataset; by the end of one *epoch*, a *neural network* will have seen every datum within the dataset once

4.1.2 Interpretation as Qualitative Support for Model Evaluation in Various Application Domains

While research into interpretation itself is relatively new, its impact has already been seen in applied deep learning contexts. A number of applied data science and AI projects that use deep learning models include a section on interpretation to qualitatively evaluate and support the model's predictions and the work's claims overall. An example of this is an approach for end-to-end neural machine translation. In the work by Johnson et al. [65], the authors present a simple and efficient way to translate between multiple languages using a single model, taking advantage of multilingual data to improve neural machine translation for all languages involved. The authors visualize an embedding of text sequences, for example, sentences from multiple languages, to support and hint at a universal interlingua representation. Another work that visualizes large machine learning embeddings is by Zahavy et al. [56], where the authors analyze deep Q-networks (DQN), a popular reinforcement learning model, to understand and describe the policies learned by DQNs for three different Atari 2600 video games. An application for social good by Robinson et al. [49] demonstrates how to apply deep neural networks on satellite imagery to perform population prediction and disaggregation, jointly answering the questions "where do people live?" and "how many people live there?". In general, they show how their methodology can be an effective tool for extracting information from inherently unstructured, remotely-sensed data to provide effective solutions to social problems.

These are only a few domains where visualization and deep learning interpretation have been successfully used. Others include building trust in autonomous driving vehicles [30], explaining decisions made by medical imaging models, such as MRIs on brain scans, to provide medical experts more information for making diagnoses [66], and using visual analytics to explore automatically-learned features from street imagery to gain perspective into identity, function, demographics, and affluence in urban spaces, which is useful for urban design and planning [67].

In this survey we will mention interpretation and explanation often, as they are the most common motivations for deep learning visualization. Later, we will discuss the different visualization techniques and visual analytics systems that focus on neural network interpretability for embeddings [51], text [32], [40], [41], quantifying interpretability [28], and many different image-based techniques stemming from the AI communities [4], [10], [68], [69], [70].

4.2 Debugging & Improving Models

Building machine learning models is an iterative design process [71], [72], [73], and developing deep neural networks is no different. While mathematical foundations have been laid, deep learning still has many open research questions. For

example, finding the exact combinations of model depth, layer width, and finely tuned hyperparameters is nontrivial. In response to this, many visual analytics systems have been proposed to help model developers build and debug their models, with the hope of expediting the iterative experimentation process to ultimately improve performance [15], [47], [52]. Oftentimes this requires monitoring models during the training phase [42], [58], identifying misclassified instances and testing a handful of well-known data instances to observe performance [29], [39], [50], and allowing a system to suggest potential directions for the model developer to explore [34]. This reason for why we wish to visualize deep learning ultimately provides better tools to speed up model development for engineers and researchers so that they can quickly identify and fix problems within a model to improve overall performance.

4.3   Comparing & Selecting Models

While certainly related to model debugging and improvement, model comparison and selection are slightly different tasks in which visualization can be useful [74], [75], [76]. Oftentimes model comparison describes the notion of choosing a single model among an ensemble of well-performing models. That is, no debugging needs to be done; all models have "learned" or have been trained semi-successfully. Therefore, the act of selecting a single, best-performing model requires inspecting model metrics and visualizing parts of the model to pick the one that has the highest accuracy, the lowest loss, or is the most generalizable, while avoiding pitfalls such as memorizing training data or overfitting.

Some systems take a high-level approach and compare user-defined model metrics, like accuracy and loss, and aggregate them on interactive charts for performance comparison [27]. Other frameworks compare neural networks trained on different random initializations (an important step in model design) to discover how they would affect performance, while also quantifying performance and interpretation [28]. Some approaches compare models on image generation techniques, such as performing image reconstruction from the internal representations of each layer of different networks to compare different network architectures [77]. Similar to comparing model architectures, some systems solely rely on data visualization representations and encodings to compare models [43], while others compare different snapshots of a single model as it trains over time, i.e., comparing a model after n_1 epochs and the same model after n_2 epochs of training time [57].

4.4   Teaching Deep Learning Concepts

Apart from AI experts, another important reason why we may wish to visualize deep learning is to educate non-expert users about AI. The exact definition of non-experts varies by source and is discussed further in Section 5.3. An example that targets the general public is Teachable Machines [54], a web-based AI experiment that explores and teaches the foundations of an image classifier. Users train a three-way image classifier by using their computer's webcam to generate the training data. After providing three different examples of physical objects around the user (e.g., holding up a pencil, a coffee mug, and a phone), the system then performs real-time inference on whichever object is in

view of the webcam, and shows a bar chart with the corresponding classification scores. Since inference is computed in real-time, the bar charts wiggles and jumps back and forth as the user removes an object, say the pencil, from the view and instead holds up the coffee mug. The visualization used is a simple bar chart, which provides an approachable introduction into image classification, a modern-day computer vision and AI problem.

Another example for teaching deep learning concepts, the Deep Visualization Toolbox [55] discussed later in this survey, also uses a webcam for instant feedback when interacting with a neural network. Taking instantaneous feedback a step further, some works have used direct manipulation to engage non-experts in the learning process. TensorFlow Playground [16], a robust, web-based visual analytics tool for exploring simple neural networks, uses direct manipulation to reinforce deep learning concepts, and importantly, evokes the user's intuition about how neural networks work. Other non-traditional mediums have been used to teach deep learning concepts and build an intuition for how neural networks behave too. Longform, interactive scrollytelling works focusing on particular AI topics that use interactive visualizations as supporting evidence are gaining popularity. Examples include "How to Use t-SNE Effectively," where users can play with hundreds of small datasets and vary single parameters to observe their effect on an embedding [78], and a similar interactive article titled "Visualizing MNIST" that visualizes different types of embeddings produced by different algorithms [45].

5   WHO USES DEEP LEARNING VISUALIZATION

This section describes the groups of people who may stand to benefit from deep learning visualization and visual analytics. We loosely organize them into three *non-mutually exclusive* groups by their level of deep learning knowledge (most to least): *model developers*, *model users*, and *non-experts*. Note that many of the works discussed can benefit multiple groups, e.g., a model developer may use a tool aimed at non-experts to reinforce their own intuition for how neural networks learn.

5.1   Model Developers & Builders

The first group of people who use deep learning visualization are individuals whose job is primarily focused on developing, experimenting with, and deploying deep neural networks. These model developers and builders, whether they are researchers or engineers, have a strong understanding of deep learning techniques and a well-developed intuition surrounding model building. Their knowledge expedites key decisions in deep learning workflows, such as identifying the which types of models perform best on which types of data. These individuals wield mastery over models, e.g., knowing how to vary hyperparameters in the right fashion to achieve better performance. These individuals are typically seasoned in building large-scale models and training them on high-performance machines to solve real-world problems [24]. Therefore, tooling and research for these users is much more technically focused, e.g., exposing many hyperparameters for detailed model control.

Of the existing deep learning visual analytics tools published, a handful tackle the problem of developing tools for

model developers, but few have seen widespread adoption. Arguably the most well-known system is TensorBoard [27]: Google's included open-source visualization platform for its dataflow graph library TensorFlow. TensorBoard includes a number of built-in components to help model developers understand, debug, and optimize TensorFlow programs. It includes real-time plotting of quantitative model metrics during training, instance-level predictions, and a visualization of the computational graph. The computational graph component was published separately by Wongsuphasawat et al. [15] and works by applying a series of graph transformations that enable standard layout techniques to produce interactive diagrams of TensorFlow models.

Other tools, such as DeepEyes [47], assist in a number of model building tasks, e.g., identifying stable layers during the training process, identifying unnecessary layers and degenerate filters that do not contribute to a model's decisions, pruning such entities, and identifying patterns undetected by the network, indicating that more filters or layers may be needed. Another tool, *Blocks* [29], allows a model builder to accelerate model convergence and alleviate overfitting, through visualizing class-level confusion patterns. Other research has developed new metrics beyond measures like loss and accuracy, to help developers inspect and evaluate networks while training them [58].

Some tools also address the inherent iterative nature of training neural networks. For example, ML-o-scope [31] utilizes a time-lapse engine to inspect a model's training dynamics to better tune hyperparameters, while work by Chae et al. [34] visualizes classification results during training and suggests potential directions to improve performance in the model building pipeline. Lastly, visual analytics tools are beginning to be built for expert users who wish to use models that are more challenging to work with. For example, DGMTracker [42] is a visual analytics tool built to help users understand and diagnose the training process of deep generative models: powerful networks that perform unsupervised and semi-supervised learning where the primary focus is to discover the hidden structure of data without resorting to external labels.

5.2 Model Users

The second group of people who may benefit from deep learning visualization are model users. These are users who may have some technical background but are neural network novices. Common tasks include using well-known neural network architectures for developing domain specific applications, training smaller-scale models, and downloading pre-trained model weights online to use as a starting point. This group of users also include machine learning artists who use models to enable and showcase new forms of artistic expression.

An example visual analytics system for these model users is ActiVis [39]: a visual analytics system for interpreting the results of neural networks by using a novel visual representation that unifies instance- and subset-level inspections of neuron activations. Model users can flexibly specify subsets using input features, labels, or any intermediate outcomes in a machine learning pipeline. ActiVis was built for engineers and data scientists at Facebook to explore and interpret deep learning models results and is deployed on

Fig. 2. ActiVis [39]: A visual analytics system or interpreting neural network results using a novel visualization that unifies instance- and subset-level inspections of neuron activations deployed at Facebook.

Facebook's internal system. LSTMVis [52] is a visual analysis tool for recurrent neural networks with a focus on understanding hidden state dynamics in sequence modeling. The tool allows model users to perform hypothesis testing by selecting an input range to focus on local state changes, then to match these states changes to similar patterns in a large dataset, and finally align the results with structural annotations. The LSTMVis work describes three types of users: architects, those who wish to develop new deep learning methodologies; trainers, those who wish to apply LSTMs to a task in which they are domain experts in; and end users, those who use pretrained models for various tasks. Lastly, Embedding Projector [51], while not specifically deep learning exclusive, is a visual analytics tool to support interactive visualization and interpretation of large-scale embeddings, which are common outputs from neural network models. The work presents three important tasks that model users often perform while using embeddings; these include exploring local neighborhoods, viewing the global geometry to find clusters, and finding meaningful directions within an embedding.

5.3 Non-Experts

The third group of people whom visualization could aid are non-experts in deep learning. These are individuals who typically have no prior knowledge about deep learning, and may or may not have a technical background. Much of the research targeted at this group is for educational purposes, trying to explain what a neural network is and how it works at a high-level, sometimes without revealing deep learning is present. These group also includes people who simply use AI-powered devices and consumer applications.

Apart from Teachable Machines [54] and the Deep Visualization Toolbox [55] mentioned in Section 4.4, TensorFlow Playground [16], a web-based interactive visualization of a simple dense network, has become a go-to tool for gaining intuition about how neural networks learn. TensorFlow Playground uses direct manipulation experimentation rather than coding, enabling users to quickly build an intuition about neural networks. The system has been used to teach students about foundational neural network properties by using "living lessons," and also makes it straightforward to create a dynamic, interactive educational

experience. Another web-browser based system, Shape-Shop [38], allows users to explore and understand the relationship between input data and a network's learned representations. ShapeShop uses a feature visualization technique called class activation maximization to visualize specific classes of an image classifier. The system allows users to interactively select classes from a collection of simple shapes, select a few hyperparameters, train a model, and view the generated visualizations all in real-time.

Tools built for non-experts, particularly with an educational focus, are becoming more popular on the web. A number of web-based JavaScript frameworks for training neural networks and inference have been developed; however, ConvNetJS (http://cs.stanford.edu/people/karpathy/convnetjs/) and TensorFlow.js (https://js.tensorflow.org/) are the most used and have enabled developers to create highly interactive explorable explanations for deep learning models.

6   WHAT TO VISUALIZE IN DEEP LEARNING

This section discusses the technical components of neural networks that could be visualized. This section is strongly related to the next section, Section 7 "How," which describes how the components of these networks are visualized in existing work. By first describing *what* may be visualized (this section), we can more easily ground our discussion on *how* to visualize them (next section).

6.1   Computational Graph & Network Architecture

The first thing that can be visualized in a deep learning model is the model architecture. This includes the *computational graph* that defines how a neural network model would train, test, save data to disk, and checkpoint after epoch iterations [27]. Also called the dataflow graph [27], this defines how data flows from operation to operation to successfully train and use a model. This is different than the neural network's edges and weights, discussed next, which are the parameters to be tweaked during training. The dataflow graph can be visualized to potentially inform model developers of the types of computations occurring within their model, as discussed in Section 7.1.

6.2   Learned Model Parameters

Other components that can be visualized are the learned parameters in the network during and after training.

6.2.1   Neural Network Edge Weights

Neural network models are built of many, and sometimes diverse, constructions of layers of computational units [26]. These layers send information throughout the network by using edges that connect layers to one another, oftentimes in a linear manner, yet some more recent architectures have shown that skipping certain layers and combining information in unique ways can lead to better performance. Regardless, each node has an outgoing edge with an accompanying *weight* that sends signal from one neuron in a layer to potentially thousands of neurons in an adjacent layer [16]. These are the parameters that are tweaked during the backpropagation phase of training a deep model, and could be worthwhile

to visualize for understanding what the model has learned, as seen in Section 7.1.

6.2.2   Convolutional Filters

Convolutional neural networks are built using a particular type of layer, aptly called the *convolutional layer*. These convolutional layers apply filters over the input data, oftentimes images represented as a two-dimensional matrix of values, to generate smaller representations of the data to pass to later layers in the network. These filters, like the previously mentioned traditional weights, are then updated throughout the training process, i.e., learned by the network, to support a given task. Therefore, visualizing the learned filters could be useful as an alternate explanation for what a model has learned [10], [55], as seen in Section 7.6.

6.3   Individual Computational Units

Albeit reductionist, neural networks can be thought as a collection of layers of neurons connected by edge weights. Above, we discussed that the edges can be visualized, but the neurons too can be a source of data to investigate.

6.3.1   Activations

When given a trained model, one can perform inference on the model using a new data instance to obtain the neural network's output, e.g., a classification or a specific predicted value. Throughout the network, the neurons compute *activations* using activation functions (e.g., weighted sum) to combine the signal from the previous layer into a new node [26], [55]. This mapping is one of the characteristics that allows a neural network to learn. During inference, we can recover the activations produced at each layer. We can use activations in multiple ways, e.g., as a collection of individual neurons, spatial positions, or channels [46]. Although these feature representations are typically high-dimensional vectors of the input data at a certain stage within the network [46], it could be valuable in helping people visualize how input data is transformed into higher-level features, as seen in Section 7.2. Feature representations may also shed light upon how the network and its components respond to particular data instances [55], commonly called instance-level observation; we will discuss this in detail in Sections 7.4 and 7.5.

6.3.2   Gradients for Error Measurement

To train a neural network, we commonly use a process known as backpropagation [26]. *Backpropagation*, or sometimes called the backpropagation of errors, is a method to calculate the gradient of a specified loss function. When used in combination with an optimization algorithm, e.g., gradient descent, we can compute the error at the output layer of a neural network and redistribute the error by updating the model weights using the computed gradient. These gradients flow over the same edges defined in the network that contain the weights, but flow in the opposite direction., e.g., from the output layer to the input layer. Therefore, it could be useful to visualize the gradients of a network to see how much error is produced at certain outputs and where it is distributed [33], [35], as mentioned in Section 7.6.

6.4 Neurons in High-Dimensional Space

Continuing the discussion of visualizing activations of a data instance, we can think of the feature vectors recovered as vectors in a high-dimensional space. Each neuron in a layer then becomes a "dimension." This shift in perspective is powerful, since we can now take advantage of high-dimensional visualization techniques to visualize extracted activations [48], [79]. Sometimes, people use neural networks simply as feature vector generators, and defer the actual task to other computational techniques, e.g., traditional machine learning models [4], [49]. In this perspective, we now can think of deep neural networks as feature generators, whose output embeddings could be worth exploring. A common technique is to use dimensionality reduction to take the space spanned by the activations and embed it into 2D or 3D for visualization purposes [48], [51], [79], as discussed in Section 7.2.

6.5 Aggregated Information

6.5.1 Groups of Instances

As mentioned earlier, instance-level activations allow one to recover the mapping from data input to a feature vector output. While this can be done for a single data instance, it can also be done on collections of instances. While at first this does not seem like a major differentiation from before, instance groups provide some unique advantages [39], [43]. For example, since instance groups by definition are composed of many instances, one can compute all the activations simultaneously. Using visualization, we can now compare these individual activations to see how similar or different they are from one another. Taking this further, with instance groups, we can now take multiple groups, potentially from differing classes, and compare how the distribution of activations from one group compares or differs from another. This aggregation of known instances into higher-level groups could be useful for uncovering the learned decision boundary in classification tasks, as seen in Sections 7.2 and 7.4.

6.5.2 Model Metrics

While instance- and group-level activations could be useful for investigating how neural networks respond to particular results a-priori, they suffer from scalability issues, since deep learning models typically wrangle large datasets. An alternative object to visualize are model metrics, including loss, accuracy, and other measures of error [27]. These summary statistics are typically computed every epoch and represented as a time series over the course of a model's training phase. Representing the state of a model through a single number, or handful of numbers, abstracts away much of the subtle and interesting features of deep neural networks; however, these metrics are key indicators for communicating how the network is progressing during the training phase [47]. For example, is the network "learning" anything at all or is it learning "too much" and is simply memorizing data causing it to overfit? Not only do these metrics describe notions of a single model's performance over time, but in the case of model comparison, these metrics become more important, as they can provide a quick and easy way to compare multiple models at once. For this reason, visualizing model metrics can be an important and powerful tool to consider for visual analytics, as discussed in Section 7.3.

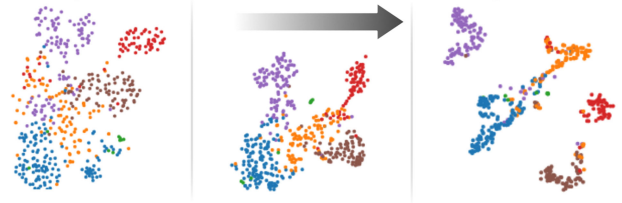Activation patterns are more discernible as data flows through the network

Fig. 3. Each point is a data instance's high-dimensional activations at a particular layer inside of a neural network, dimensionally reduced, and plotted in 2D. Notice as the data flows through the network the activation patterns become more discernible (left to right) [39].

7 HOW TO VISUALIZE DEEP LEARNING

In the previous section, we described what technical components of neural networks could be visualized. In this section, we summarize how the components are visualized and interacted with in existing literature. For most neural network components, they are often visualized using a few common approaches. For example, network architectures are often represented as node-link diagrams; embeddings of many activations are typically represented as scatter plots; and model metrics over epoch time are almost always represented as line charts. In this section, we will also discuss other representations, going beyond the typical approaches.

7.1 Node-Link Diagrams for Network Architectures

Given a neural network's dataflow graph or model architecture, the most common way to visualize where data flows and the magnitude of edge weights is a node-link diagram. Neurons are shown as nodes, and edge weights as links. For computational and dataflow graphs, Kahng et al. [39] describe two methods for creating node-link diagrams. The first represents only operations as nodes, while the second represents both operations and data as nodes. The first way is becoming the standard due to the popularity of Tensor-Board [27] and the inclusion of its interactive dataflow graph visualization [15]. However, displaying large numbers of links from complex models can generate "hairball" visualizations where many edge crossings impede pattern discovery. To address this problem, Wongsuphasawat et al. [15] extracts high-degree nodes (responsible for many of the edge crossings), visualizes them separately from the main graph, and allow users to define super-groups within the code. Another approach to reduce clutter is to place more information on each node; DGMTracker [42] provides a quick snapshot of the dataflow in and out of a node by visualizing its activations within each node.

Regarding neural network architecture, many visual analytics systems use node-link diagrams (neurons as nodes, weights as links) [13], [14], [16], [35], [37]. The weight magnitude and sign can then be encoded using color or link thickness. This technique was one of the the first to be proposed [13], and the trend has continued on in literature. Building on this technique, Harley [37] visualizes the convolution windows on each layer and how the activations propagate through the network to make the final classification. Similar to the dataflow graph examples above, some works include richer information inside each node besides an activation value, such as showing a list of images that highly

activate that neuron or the activations at a neuron as a matrix [14]. As mentioned in the dataflow graph visualizations, node-link diagrams for network architecture work well for smaller networks [16], but they also suffer from scalabilty issues. CNNVis [14], a visual analytics system that visualizes convolutional neural networks, proposes to use a bi-clustering-based edge bundling technique to reduce visual clutter caused by too many links.

7.2 Dimensionality Reduction & Scatter Plots

In Section 6, "What," we discussed different types of high-dimensional embeddings: text can be represented as vectors in word embeddings for natural language processing and images can be represented as feature vectors inside of a neural network. Both of these types of embeddings are mathematically represented as large tensors, or sometimes as 2D matrices, where each row may correspond to an instance and each column a feature.

The most common technique to visualize these embeddings is performing dimensionality reduction to reduce the number of columns (e.g., features) to two or three. Projecting onto two dimensions would mean computing (x, y) coordinates for every data instance; for three dimensions, we compute an additional z component, resulting in (x, y, z). In the 2D case, we can plot all data instances as points in a scatter plot where the axes may or may not have interpretable meaning, depending on the reduction technique used, e.g., principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) [79]. In the 3D case, we can still plot each data instance as a point in 3D space and use interactions to pan, rotate, and navigate this space [51]. These types of embeddings are often included in visual analytics systems as one of the main views [35], [47], and are also used in application papers as static Figures [56], [65]. However, viewing a 3D space on a 2D medium (e.g., computer screen) may not be ideal for tasks like comparing exact distances.

Since each reduced point corresponds to an original data instance, another common approach is to retrieve the original image and place it at the reduced coordinate location. Although the image size must be greatly reduced to prevent excessive overlap, viewing all the images at once can provide insight into what a deep learning model has learned, as seen in the example in [77] where the authors visualize ImageNet test data, or in [80] where the authors create many synthetic images from a single class and compare the variance across many random initial starting seeds for the generation algorithm. We have discussed the typical case where each dot in the scatter plot is a data instance, but some work has also visualized neurons in a layer as separate data instances [58]. Another work studies closely how data instances are transformed as their information is passed through the deep network, which in effect visualizes how the neural network separates various classes along approximated decision boundaries [48]. It is also possible to use time-dependent data and visualize how an embedding changes over time, or in the case of deep learning, over epochs [81]. This can be useful for evaluating the quality of the embedding during the training phase.

However, these scatter plots raise problems too. The quality of the embeddings greatly depends on the algorithm used to perform the reduction. Some works have studied how PCA and t-SNE differ, mathematical and visually, and suggest new reduction techniques to capture the semantic and syntactic qualities within word embeddings [82]. It has also been shown that popular reduction techniques like t-SNE are sensitive to changes in the hyperparameter space. Wattenberg meticulously explores the hyperparameter space for t-SNE, and offers lessons learned and practical advice for those who wish to use dimensionality reduction methods [78]. While these techniques are commonplace, there are still iterative improvements that can be done using clever interaction design, such as finding instances similar to a target instance, i.e., those "near" the target in the projected space, helping people build intuition for how data is spatially arranged [51].

7.3 Line Charts for Temporal Metrics

Model developers track the progression of their deep learning models by monitoring and observing a number of different metrics computed after each epoch, including the loss, accuracy, and different measure of errors. This can be useful for diagnosing the long training process of deep learning models., The most common visualization technique for visualizing this data is by considering the metrics as time series and plotting them in line charts [27]. This approach is widely used in deep learning visual analytics tools [35], [47]. After each epoch, a new entry in the time series is computed, therefore some tools, like TensorBoard, run alongside models as they train and update with the latest status [27]. TensorBoard focuses much of its screen real-estate to these types of charts and supports interactions for plotting multiple metrics in small multiples, plotting multiple models on top of one another, filtering different models, providing tooltips for the exact metric values, and resizing charts for closer inspection. This technique appears in many visual analytics systems and has become a staple for model training, comparison, and selection.

7.4 Instance-Based Analysis & Exploration

Another technique to help interpret and debug deep learning models is testing specific data instances to understand how they progress throughout a model. Many experts have built up their own collection of data instances over time, having developed deep knowledge about their expected behaviors in models while also knowing their ground truth labels [19], [39]. For example, an instance consisting of a single image or a single text phrase is much easier to understand than an entire image dataset or word embedding consisting of thousands of numerical features extracted from an end user's data. This is called instance-level observation, where intensive analysis and scrutiny is placed on a single data instance's transformation process throughout the network, and ultimately its final output.

7.4.1 Identifying & Analyzing Misclassified Instances

One application of instance-level analysis is using instances as unit tests for deep learning models. In the best case scenario, all the familiar instances are classified or predicted correctly; however, it is important to understand *when* a specific instance can fail and *how* it fails. For example, in the

task of predicting population from satellite imagery, the authors showcase three maps of areas with high errors by using a translucent heatmap overlaid on the satellite imagery [49]. Inspecting these instances reveals three geographic areas that contain high amounts of man-made features and signs of activity but have no registered people living in them: an army base, a national lab, and Walt Disney World. The visualization helps demonstrate that the proposed model is indeed learning high-level features about the input data. Another technique, HOGgles [83], uses an algorithm to visualize feature spaces by using object detectors while inverting visual features back to natural images. The authors find that when visualizing the features of misclassified images, although the classification is wrong in the image space, they look deceptively similar to the true positives in the feature space. Therefore, by visualizing feature spaces of misclassified instances, we can gain a more intuitive understanding of recognition systems.

For textual data, a popular technique for analyzing particular data instances is to use color as the primary encoding. For example, the background of particular characters in a phrase of words in a sentence would be colored using a divergent color scheme according to some criteria, often their activation magnitudes [32], [36], [40]. This helps identify particular data instances that may warrant deeper inspection (e.g., those misclassified) [19].

When pre-defined data instances are not unavailable (e.g., when analyzing a new dataset), how can we guide users towards important and interesting instances? To address this problem, a visual analytics system called *Blocks* [29] uses confusion matrices, a technique for summarizing the performance of a classification algorithm, and matrix-level sorting interactions to reveal that class error often occurs in hierarchies. *Blocks* incorporates these techniques with a sample viewer in the user interface to show selected samples potentially worth exploring.

7.4.2 Analyzing Groups of Instances

Instead of using individual data instances for testing and debugging a model, it is also common for experts to perform similar similar tasks using groups of instances [19]. While some detail may be lost when performing group-level analysis it allows experts to further test the model by evaluating its average and aggregate performance across different groups.

Much of the work using this technique is done on text data using LSTM models [52]. Some approaches compute the saliency for groups of words across the model and visualize the values as a matrix [41], while others use matrix visualizations to show the activations of word groups when represented as feature vectors in word embeddings [50], [84]. One system, ActiVis [39], places instance group analysis at the focus of its interactive interface, allowing users to compare preset and user-defined groups of activations. Similar to the matrix visualization that summarizes activations for each class in CNNVis [14], ActiVis also uses a scrolling matrix visualization to unify both instance-level and group-level analysis into a single view where users can compare the activations of the user-defined instances.

However, sometimes it can be challenging to define groups for images or text. For textual data, people often use words to group documents and provide aggregated data. ConceptVector [85] addresses the instance group generation problem by providing an interactive interface to create interesting groups of concepts for model testing. Furthermore, this system also suggests additional words to include in the user-defined groups, helping guide the user to create semantically sound concepts.

7.5 Interactive Experimentation

Interactive experimentation, another interesting area that integrates deep learning visualization, makes heavy use of interactions for users to experiment with models [86]. By using direct manipulation for testing models, a user can pose "what if?" questions and observe how the input data affects the results. Called *explorable explanations* [87], this type of visual experimentation is popular for making sense of complex concepts and systems.

7.5.1 Models Responding to User-Provided Input Data

To engage the user with the desired concepts to be taught, many systems require the user to provide some kind of input data into the system to obtain results. Some visual analytics systems use a webcam to capture live videos, and visualize how the internals of neural network models respond to these dynamic inputs [55]. Another example is a 3D visualization of a CNN trained on the classic MNIST dataset[1] that shows the convolution windows and activations on images that the user draws by hand [37]. For example, drawing a "5" in the designated area passes that example throughout the network and populates the visualization with the corresponding activations using a node-link diagram. A final example using image data is ShapeShop [38], a system that allows a user to select data from a bank of simple shapes to be classified. The system then trains a neural network and using the class activation maximization technique to generate visualizations of the learned features of the model. This can be done in real-time, therefore a user can quickly train multiple models with different shapes to observe the effect of adding more diverse data to improve the internal model representation.

An example using textual data is the online, interactive Distill article for handwriting prediction [32], which allows a user to write words on screen, and in real-time, the system draws multiple to-be-drawn curves predicting what the user's next stroke would be, while also visualizing the model's activations. Another system uses GANs to interactively generate images based off of user's sketches [59]. By sketching a few colored lines, the system presents the user with multiple synthetic images using the sketch as a guideline for what to generate. A final example is the Adversarial Playground [44], a visual analytics system that enables users to compare adversarially-perturbed images, to help users understand why an adversarial example can fool a CNN image classifier. The user can select from one of the MNIST digits and adjust the strength of adversarial attack. The system then compares the classifications scores in a bar chart to

---

1. MNIST is a small, popular dataset consisting of thousands of 28×28px images of handwritten digits (0 to 9). MNIST is commonly used as a benchmark for image classification models.
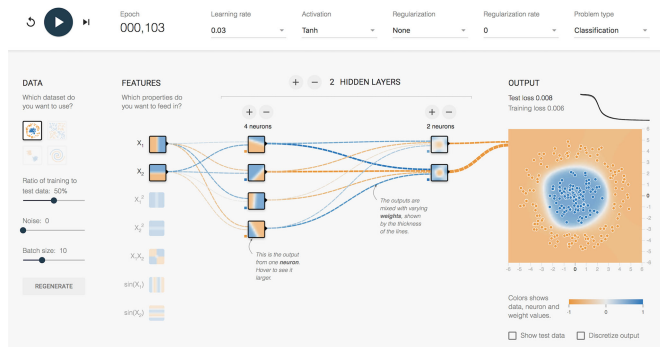
Fig. 4. TensorFlow Playground [16]: A web-based visual analytics tool for exploring simple neural networks that uses direct manipulation rather than programming to teach deep learning concepts and develop an intuition about how neural networks behave.

observe how simple perturbations can greatly impact classification accuracy.

7.5.2 How Hyperparameters Affect Results

While deep learning models automatically adjust their internal parameters, their hyperparameters still require fine-tuning. These hyperparameters can have major impact on model performance and robustness. Some visual analytics systems expose model hyperparameters to the user for interactive experimentation. One example previously mentioned is TensorFlow Playground [16], where users can use direct manipulation to adjust the architecture of a simple, fully-connected neural network, as well as the hyperparameters associated with its training, such as the learning rate, activation function, and regularization. Another example is a Distill article that meticulously explores the hyperparaemters of the t-SNE dimensionality reduction method [78]. This article tests dozens of synthetic datasets in different arrangements, while varying hyperparameters such as the t-SNE perplexity and the number of iterations to run the algorithm for.

7.6 Algorithms for Attribution & Feature Visualization

The final method for how to visualize deep learning hails from the AI and computer vision communities. These are algorithmic techniques that entail image generation. Given a trained a model, one can select a single image instance and use one of the algorithmic techniques to generate a new image of the same size that either highlights important regions of the image (often called *attribution*) or is an entirely new image that supposedly is representative of the same class (often called *feature visualization*) [46], [88]. In these works, it is common to see large, full-page figures consisting of hundreds of such images corresponding to multiple images classes [89]. However, it is uncommon to see interactivity in these works, as the primary contribution is often about algorithms, not interactive techniques or systems. Since the focus of this interrogative survey is on visual analytics in deep learning, we do not discuss in detail the various types of algorithmic techniques. Rather, we mention the most prominent techniques developed, since they are impactful to the growing field of deep learning visualization and could be incorporated into visual analytics systems in the future. For more details about these techniques, such as

input modification, deconvolutional methods [10], and input reconstruction methods, we refer our readers to the taxonomies [90] and literature surveys for visualizing learned features in CNNs [22], [91], and a tutorial that presents the theory behind many of these interpretation techniques and discusses tricks and recommendations to efficiently use them on real data [61].

7.6.1 Heatmaps for Attribution, Attention, & Saliency

One research area generates translucent heatmaps that overlay images to highlight important regions that contribute towards classification and their sensitivity [4], [69], [92], [93], [94]. One technique called visual backpropagation attempts to visualize which parts of an image have contributed to the classification, and can do so in real-time in a model debugging tool for self-driving vehicles [30]. Another technique is to invert representations, i.e., attempt to reconstruct an image using a feature vector to understand the what a CNN has learned [77], [95], [96]. Prediction difference analysis is a method that highlights features in an image to provide evidence for or against a certain class[66]. Other work hearkens back to more traditional computer vision techniques by exploring how object detectors emerge in CNNs and attempts to give humans object detector vision capabilities to better align humans and deep learning vision for images [83], [97]. Visualizing CNN filters is also popular, and has famously shown to generate dream-like images, becoming popular in artistic tasks [98], [99]. Some work for interpreting *visual question answering (VQA)* models and tasks use these heatmaps to explain which parts of an image a VQA model is looking at in unison with text activation maps when answering the given textual questions [36]. However, recent work has shown that some of these methods fail to provide correct results and argue that we should develop explanation methods that work on simpler models before extending them to the more complex ones [91].

7.6.2 Feature Visualization

For feature visualization, while some techniques have proven interesting [100], one of the most studied techniques, class activation maximization, maximizes the activation of a chosen, specific neuron using an optimization scheme, such as gradient ascent, and generates synthetic images that are representative of what the model has learned about the chosen class [68]. This led to a number of works improving the quality of the generated images. Some studies generated hundreds of these non-deterministic synthetic images and clustered them to see how variations in the class activation maximization algorithm affects the output image [80]. In some of their most recent work on this topic, Ngyuen et al. [70] present hundreds of high-quality images using a deep generator network to improve upon the state-of-the-art, and include figures comparing their technique to many of the existing and previous attempts to improve the quality of generated images. The techniques developed in this research area have improved dramatically over the past few years, where now it is possibly to synthetically generate photorealistic images [101]. A recent comparison of feature visualization techniques highlights their usefulness [88]; however, the authors note that they remain skeptical of their trustworthiness, e.g., do neurons have a

consistent meaning across different inputs, and if so, is that meaning accurately reified by feature visualization [46]?

8 WHEN TO VISUALIZE IN THE DEEP LEARNING PROCESS

This section describes when visualizing deep learning may be most relevant and useful. Our discussion primarily centers around the training process: an iterative, foundational procedure for using deep learning models. We identify two distinct, non-mutually exclusive times for when to visualize: *during training* and *after training*. Some works propose that visualization be used both during and after training.

8.1 During Training

Artificial neural networks learn higher-level features that are useful for class discrimination as training progress [102]. By using visualization during the training process, there is potential to monitor one's model as it learns to closely observe and track the model's performance [48].

Many of the systems in this category run in a separate web-browser alongside the training process, and interface with the underlying model to query the latest model status. This way, users can visually explore and rigorously monitor their models in real time, while they are trained elsewhere. The visualization systems dynamically update the charts with metrics recomputed after every epoch, e.g., the loss, accuracy, and training time. Such metrics are important to model developers because they rely on them to determine if a model (1) has begun to learn anything at all; (2) is converging and reaching the peak of its performance; or (3) has potentially overfitted and memorized the training data. Therefore, many of the visual analytics systems used during training support and show these updating visualizations as a primary view in the interface [16], [27], [34], [35], [42], [47]. One system, Deep View [58], visualizes model metrics during the training process and uses its own defined metrics for monitoring (rather than the loss): a discriminability metric, which evaluates neuron evolution, and a density metric which evaluates the output feature maps. This way, for detecting overfitting, the user does not need to wait long to view to infer overfitting; they simply observe the neuron density early in training phase.

Similarly, some systems help reduce development time and save computational resources by visualizing metrics that indicate whether a model is successfully learning or not, allowing a user to stop the training process early [16]. By using visualization during model training, users can save development time through model steering [35] and utilizing suggestions for model improvement [34]. Lastly, another model development time minimization focuses on diagnosing neurons and layers that are not training correctly or are misclassifying data instances. Examples include Deep-Eyes [47], a system that identifies stable and unstable layers and neurons so users may prune their models to speed up training; *Blocks* [29], a system that visualizes class confusion and reveals that confusion patterns follow a hierarchical structure over the classes which can then be exploited to design hierarchy-aware architectures; and DGMTracker [42], a system that proposes a credit assignment algorithm that

indicates how other neurons contribute to the output of particular failing neurons.

8.2 After Training

While some works support neural network design during the iterative model building process, there are other works that focus their visualization efforts after a model has been trained. In other words, these works assume a trained model as input to the system or visualization technique. Note that many, if not most, of the previously mentioned algorithmic techniques developed in the AI fields, such as attribution and feature visualization, are performed after training. These techniques are discussed more in Section 7.6.

The Embedding Projector [51] specializes in visualizing 2D and 3D embeddings produced by trained neural networks. While users can visualize typical high-dimensional datasets in this tool, the Embedding Projector tailors the experience towards embeddings commonly used deep learning. Once a neural network model has been trained, one can compute the activations for a given test dataset and visualize the activations in the Embedding Projector to visualize and explore the space that the network has learned. Instead of generating an overview embedding, another previously discussed system, the Deep Visualization Toolbox [55], uses a trained model to visualize live activations in a large small-multiples view to understand of what types of filters a convolutional network has learned.

More traditional visual analytics systems have also been developed to inspect a model after it has finished training. ActiVis [39], a visual analytics system for neural network interpretation deployed at Facebook reports that Facebook engineers and data scientists use visual analytics systems often in their normal workflow. Another system, RNNVis [43], visualizes and compares different RNN models for various natural language processing tasks. This system positions itself as a natural extension of TensorFlow; using multiple Tensor-Flow models as input, the system then analyzes the trained models to extract learned representations in hidden states, and further processes the evaluation results for visualization. Lastly, the LSTMVis [52] system, a visual analysis tool for RNN interpretability, separates model training from the visualization. This system takes a model as input that must be trained separately, and from the model, gathers the required information to produce the interactive visualizations to be rendered in a web-based front-end.

9 WHERE IS DEEP LEARNING VISUALIZATION

For the last question of the interrogative survey, we divide up "Where" into two sections: where deep learning visualization research has been applied, and where deep learning visualization research has been conducted, describing the new and hybrid community. This division provides a concise summary for practitioners who wish to investigate the usage of the described techniques for their own work, and provides new researchers with the main venues for this research area to investigate existing literature.

9.1 Application Domains & Models

While many non-neural approaches are used for real-world applications, deep learning has successfully achieved state-

of-the-art performance in several domains. Previously in Section 4.1.2, we presented works that apply neural networks to particular domains and use visualizations to lend qualitative support to their usual quantitative results to strengthen users' trust in their models. These domains included neural machine translation [65], reinforcement learning [56], social good [49], autonomous vehicles [30], medical imaging diagnostics [66], and urban planning [67].

Next we summarize the types of models that have been used in deep learning visualization. Much of the existing work has used image-based data and models, namely CNNs, to generate attribution and feature visualization explanations for what a model has learned from an image dataset. CNNs, while not exclusively used for images, have become popular in the computer vision community and are often used for image classification and interactive, image-based creative tasks [59], [103]. Besides images, sequential data (e.g., text, time series data, and music) has also been studied. This research stems from the natural language processing community, where researchers typically favor RNNs for learning representations of large text corpora. These researchers make sense of large word embeddings by using interactive tools that support dimensionality reduction techniques to solve problems such as sequence-to-sequence conversion, translation, and audio recognition. Research combining both image and text data has also been done, such as image captioning and visual question answering [104], [105]. Harder still are new types of networks called *generative adversarial networks*, or GANs for short, that have produced remarkable results for data generation [106], e.g., producing real-looking yet fake images [107]. While GANs have only existed for a couple of years, they are now receiving significant research attention. To make sense of the learned features and distributions from GANs, two visual analytics systems, DGMTracker [42] and GANViz [53], focus on understanding the training dynamics of GANs to help model developers better train these complex models, often consisting of multiple dueling neural networks.

9.2 A Vibrant Research Community: Hybrid, Apace, & Open-Sourced

As seen from this survey, bringing together the visualization communities with the AI communities has led to the design and development of numerous tools and techniques for improving deep learning interpretability and democratization. This hybrid research area has seen accelerated attention and interest due to its widespread impact, as evidenced by the large number of works published in just a few years, as seen in Table 2. A consequence of this rapid progress is that deep learning visualization research are being disseminated across multiple related venues. In academia, the premiere venues for deep learning visualization research consists of two main groups: the information visualization and visual analytics communities; and the artificial intelligence and deep learning communities. Furthermore, since this area is relatively new, it has seen more attention at multiple workshops at the previously mentioned academic conferences, as tabulated in Table 1.

Another consequence of this rapidly developing area is that new work is immediately publicized and open-sourced, without waiting for it to be "officially" published at conferences, journals, etc. Many of these releases take the form of a

preprint publication posted on arXiv, where a deep learning presence has thrived. Not only is it common for academic research labs and individuals to publish work on arXiv, but companies from industry are also publishing results, code, and tools. For example, the most popular libraries[2] for implementing neural networks are open-source and have consistent contributions for improving all areas of the codebase, e.g., installation, computation, and deployment into specific programming languages' open-source environments.

Some works have a corresponding blog post on an industry research blog[3], which, while non-traditional, has large impact due to their prominent visibility and large readership. While posting preprints may have its downsides (e.g., little quality control) the communities have been promoting the good practices of open-sourcing developed code and including direct links within the preprints; both practices are now the norm. Although it may be overwhelming to digest the amount of new research published daily, having access to the work with its code could encourage reproducibility and allow the communities to progress faster. In summary, given the increasing interest in deep learning visualization research and its importance, we believe our communities will continue to thrive, and will positively impact many domains for years to come.

10 RESEARCH DIRECTIONS & OPEN PROBLEMS

Now we present research directions and open problems for future research distilled from the surveyed works.

10.1 Furthering Interpretability

Unsurprisingly, with the amount of attention and importance on interpretability and explainability in the deep learning visualization literature, the first area for future work is continuing to create new interpretable methods for deep learning models. For the information visualization and visual analytics communities, this could constitute creating new visual representations for the components in deep learning models, or developing new interaction techniques to reveal deeper insights about one's model. For the AI communities, more insightful attribution and feature visualization techniques for trained models that are fast (computationally cheap) could be incorporated into visualization systems. Combining visual representations, helpful interactions, and state-of-the-art attribution and feature visualization techniques together into rich user interfaces could lead to major breakthroughs for understanding neural networks [46].

10.2 System & Visual Scalability

Throughout this survey, we have covered many visual analytics systems that facilitate interpretation and model understanding. However, some systems suffer from scalability problems. Visual scalability challenges arise when dealing with large data, e.g., large number of hyperparameters and millions of parameters in deep neural networks. Some research has started to address this, by simplifying

---

2. Popular libraries include TensorFlow [27], Keras, Caffe, PyTorch, and Theano.

3. High impact industry blogs include: Google Research Blog, OpenAI, Facebook Research Blog, the Apple Machine Learning Journal, NVIDIA Deep Learning AI, and Uber AI.
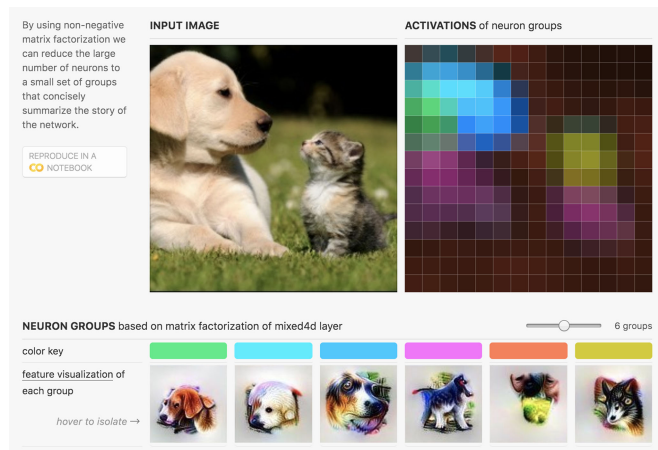
Fig. 5. Distill: The Building Blocks of Interpretability [46]: An interactive user interface that combines feature visualization and attribution techniques to interpret neural networks.

complex dataflow graphs and network weights for better model explanations [14], [15], [52]. But, regarding activations and embeddings, dimensionality reduction techniques have a limit to their usability when it comes to the number of points to visualize [48]. We think this is an important research direction, especially given that the information visualization communities have developed techniques to visualize large, high-dimensional data that could potentially be applicable to deep learning [108].

Aside from visual scalability, some tools also suffer from system scalability. While some of these problems may be more engineering-centric, we think that for visual analytics systems to adopted, they need to handle state-of-the-art deep models without penalizing performance or increasing model development time. Furthermore, these systems (often web-based) will greatly benefit from fast computations, supporting real-time, rich user interactions [50]. This is especially important for visual systems that need to perform pre-computation before rendering visualizations to the screen.

10.3 Design Studies for Evaluation: Utility & Usability

An important facet of visualization research is the evaluation of the utility and usefulness of the visual representation. Equally important is to evaluate the usability of deployed systems and their interactive visual analytics techniques. It is encouraging to see many of the visual analytics systems recognize this importance and report on design studies conducted with AI experts before building a tool to understand the users and their needs [14], [15], [19], [39], [43], [52]. It is common to see example use cases or illustrative usage scenarios that demonstrate the capabilities of the interactive systems. Some works go beyond these and conduct user studies to evaluate utility and usability [31]. In the AI communities, most works do not include user studies. For those that do, they greatly benefit from showing why their proposed methods are superior to the ones being tested against [69], [89], [109], [110]. Taking this idea to the quantifiable extreme, a related avenue of evaluating these techniques is the notion of quantifying interpretability, which has been recently studied [28], [111]. Other domains have recognized that interpretable deep learning research may require evaluation

techniques for their interpretations, and argue that there is a large body of work from fields such as philosophy, cognitive science, and social psychology that could be utilized [62], [112].

When surveying the interfaces of deep learning visual analytics tools, many of them contain multiple-coordinated views with many visual representations. Displaying this much information at once can be overwhelming, and when interpretability is the primary focus, it is critical for these systems to have superior usability. Therefore, we think future works could further benefit from including more members of the human-computer interaction communities, including interface and user experience designers, that could help organize and prioritize interfaces using well-studied guidelines [86].

10.4 The Human Role in Interpretability

10.4.1 Human v. Machine Understanding of the World

In deep learning interpretability work, researchers are developing methods to produce better explanations to "see through the black-box," but unfortunately some of these methods produce visualizations that, while visually interesting and thought-provoking [98], are not fully understandable by their human viewers. That is an important facet of deep learning interpretability, namely, producing visualizations and interpretations that are human understandable [88]. Some methods compare algorithmic results with an empirically derived human baseline; this enables comparison between machine and human generated responses to objects in the world, particularly in images [113]. Ultimately, researchers seek to understand the commonalities and differences between how humans and machines see and decompose the world [114]. Some tools that we have surveyed achieve this by using live-video to compare the input images and the neural network's activations and filters in real time [55]. Other tools give users explicit control of an experiment by training multiple small models with only a few exposed hyperparameters, automatically generating visualizations to then see the effect that the input data has on the learned representation [38]. These "what-if" tools and scenarios could potentially be extended to incorporate human feedback into the training or model steering process of neural network to better improve performance.

10.4.2 Human-AI Pairing

Much of this survey is dedicated towards reviewing the state-of-the-art in visual analytics for deep learning, with a focus on interpretability. These works use visualization to explain, explore, and debug models in order to choose the best preforming model for a given task, often by placing a human in the loop. However, a slight twist on this idea hearkening back to the original envisioning of the computer has lead to the emergence of a new research area, one where tasks are not exclusively performed by humans or machines, but one where the two complement each other. This area, recently dubbed *artificial intelligence augmentation* describes the use of AI systems to help develop new methods for intelligence augmentation [103]. Some related works we have covered already propose artificial intelligence augmentation ideas, such as a system that suggests potentially interesting directions to explore in a high-dimensional 3D

embedding [51], predicting and showing where the next stroke of a word could be when handwriting text [32], automatically generating images based off of user-provided sketches [59], and dynamically changing and steering a neural network model while it trains [35]. We believe this is a rich, under-explored area for future research: using well-designed interfaces for humans to interact with machine learning models, and for these machine learning models to augment creative human tasks.

10.5   Social Good & Bias Detection

The aspirational pairing of humans and machines is a long-term research endeavor. To quicken our pace, we must continue to democratize artificial intelligence via educational tools, perhaps by using direct manipulation as an invitation for people to engage with AI [16], [46], clear explanations for model decision making, and robust tooling and libraries for programming languages for people to develop such models [15], [27]. While doing this, we must also ensure that AI applications remain ethical, fair, safe, transparent, and are benefiting society [63].

Another important consideration for future research is detecting bias. This has been identified as a major problem in deep learning [115], [116], and a number of researchers are using visualization to understand why a model may be biased [117]. One example that aims to detect data bias is Google's Facets tool [118], a visual analytics system designed specifically to preview and visualize machine learning datasets before training. This allows one to inspect large datasets by exploring the different classes or data instances, to see if there are any high-level imbalances in the class or data distribution.

Other works have begun to explore if the mathematical algorithms themselves can be biased towards particular decisions. An example of this is an interactive article titled "Attacking discrimination with smarter machine learning" [117], which explores how one can can create both fair and unfair threshold classifiers in an example task such as loan granting scenarios where a bank may grant or deny a loan based on a single, automatically computed number such as a credit score. The article aims to highlight that *equal opportunity* [119] is not preserved by machine learning algorithms, and that as AI-powered systems continue to make important decisions across core social domains, it is critical to ensure decisions are not discriminatory.

Finally, aside from data and model bias, humans are often inherently biased decision makers. In response, there is a growing area of research into detecting and understanding bias in visual analytics[4] and its affect on the decision making process [120]. Some work has developed metrics to detect types of bias to present to a user during data analysis [120] which could also be applied to visual tools for deep learning in the future. Some work has employed developmental and cognitive psychology analysis techniques to understand how humans learn, focusing on uncovering how human bias is developed and influences learning, to ultimately influence artificial neural network design [112].

---

4. The DECISIVe Workshop (http://decisive-workshop.dbvis.de/) at IEEE VIS is dedicated to understanding cognitive bias in visualization.

10.6   Protecting Against Adversarial Attacks

Regardless of the benefits AI systems are bringing to society, we would be remiss to immediately trust them; like most technologies, AI too has security faults. Identified and studied in seminal works, it has been shown that deep learning models such as image classifiers can be easily fooled by perturbing an input image [121], [122], [123]. Most alarming, some perturbations are so subtle that they are untraceable by the human eye, yet would completely fool a model into misclassification [122]. This sparked great interest in the AI communities, and much work has been done to understand how fragile deep neural network image classifiers are, identify in what ways can they break, and explore methods for protecting them. Norton et al. [44] demonstrate adding adversarial perturbations to images in an interactive tool, where users can tweak the type and intensity of the attack, and observe the resulting (mis) classification. This is a great first start for using visualization to identify potential attacks, but we think visualization can be majorly impactful in this research space, by not only showcasing how the attacks work and detecting them, but also by taking action and protecting AI systems from the attacks themselves. While some work, primarily originating from the AI communities, has proposed computational techniques to protect AI from attacks, such as identifying adversarial examples before classification [124], modifying the network architecture [125], modifying the training process [122], [126], and performing pre-processing steps before classification [127], [128], we think visualization can have great impact for combating adversarial machine learning.

11   CONCLUSION

We presented a comprehensive, timely survey on visualization and visual analytics in deep learning research, using a human-centered, interrogative framework. Our method helps researchers and practitioners in visual analytics and deep learning to quickly learn key aspects of this young and rapidly growing body of research, whose impact spans a broad range of domains. Our survey goes beyond visualization-focused venues to extend a wide scope that also encompasses relevant works from top venues in AI, ML, and computer vision. We highlighted visual analytics as an integral component in addressing pressing issues in modern AI, helping to discover and communicate insight, from discerning model bias, understanding models, to promoting AI safety. We concluded by highlighting impactful research directions and open problems.

REFERENCES

[1]   W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin Math. Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
[2]   W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[4] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," arXiv:1312.6034, 2013.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[6] A. Karpathy, "What I learned from competing against a convnet on ImageNet," 2014. [Online]. Available: http://karpathy.github.io/2014/09/02/what-i-learned-from-eompeting-against-a-convnet-on-imagenet

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[10] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[11] M. W. Craven and J. W. Shavlik, "Visualizing learning and computation in artificial neural networks," *Int. J. Artif. Intell. Tools*, vol. 1, no. 3, pp. 399–425, 1992.

[12] M. J. Streeter, M. O. Ward, and S. A. Alvarez, "NVIS: An interactive visualization tool for neural networks," *Visual Data Exploration Anal. VIII*, vol. 4302, pp. 234–241, 2001.

[13] F.-Y. Tzeng and K.-L. Ma, "Opening the black box: Data driven visualization of neural networks," in *Proc. IEEE Vis.*, 2005, pp. 383–390.

[14] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 91–100, Jan. 2017.

[15] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg, "Visualizing dataflow graphs of deep learning models in TensorFlow," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 1–12, Jan. 2018.

[16] D. Smilkov, S. Carter, D. Sculley, F. B. Viegas, and M. Wattenberg, "Direct-manipulation visualization of deep networks," in *Proc. ICML Workshop Vis Deep Learn.*, 2016.

[17] J. Lu, W. Chen, Y. Ma, J. Ke, Z. Li, F. Zhang, and R. Maciejewski, "Recent progress and trends in predictive visual analytics," *Frontiers Comput. Sci.*, vol. 11, pp. 192–207, 2017.

[18] Y. Lu, R. Garcia, B. Hansen, M. Gleicher, and R. Maciejewski, "The state-of-the-art in predictive visual analytics," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 539–562, 2017.

[19] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting interactive performance analysis for multiclass classifiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 61–70, Jan. 2017.

[20] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Mag.*, vol. 35, no. 4, pp. 105–120, 2014.

[21] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, D. Weiskopf, S. North, and D. Keim, "Human-centered machine learning through interactive visualization," in *Proc. Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, 2016, pp. 641–646.

[22] C. Seifert, A. Aamir, A. Balagopalan, D. Jain, A. Sharma, S. Grottel, and S. Gumhold, "Visualizations of deep neural networks in computer vision: A survey," in *Transparent Data Mining for Big and Small Data*. Berlin, Germany: Springer, 2017.

[23] H. Zeng, "Towards better understanding of deep learning with visualization," The Hong Kong University of Science and Technology, PhD Qualifying Exam, 2016.

[24] S. Liu, X. Wang, M. Liu, and J. Zhu, "Towards better analysis of machine learning models: A visual analytics perspective," *Visual Informat.*, vol. 1, no. 1, 2017, pp. 48–56.

[25] J. Choo and S. Liu, "Visual analytics for explainable deep learning," *IEEE Comput. Graph. Appl.*, 2018.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, [Online]. Available: http://www.deeplearningbook.org

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," arXiv:1603.04467, 2016.

[28] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3319–3327.

[29] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren, "Do convolutional neural networks learn class hierarchy?" *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 152–162, Jan. 2018.

[30] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller, and K. Zieba, "Visualbackprop: Visualizing cnns for autonomous driving," arXiv:1611.05418, 2016.

[31] D. Bruckner, "Ml-o-scope: a diagnostic visualization system for deep machine learning pipelines," Master's thesis, EECS Dept., Univ. California, Berkeley, May 2014. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014–99.html

[32] S. Carter, D. Ha, I. Johnson, and C. Olah, "Experiments in handwriting with a neural network," *Distill*, 2016.

[33] D. Cashman, G. Patterson, A. Mosca, and R. Chang, "RNNbow: Visualizing learning via backpropagation gradients in recurrent neural networks," in *Proc. Workshop Visual Analytics Deep Learn.*, 2017.

[34] J. Chae, S. Gao, A. Ramanthan, C. Steed, and G. D. Tourassi, "Visualization for classification in deep neural networks," in *Proc. Workshop Visual Analytics Deep Learn.*, 2017.

[35] S. Chung, C. Park, S. Suh, K. Kang, J. Choo, and B. C. Kwon, "ReVACNN: Steering convolutional neural network via real-time visual analytics," in *Proc. Neural Inf. Process. Syst. Workshop Future Interactive Learn. Mach.*, 2016.

[36] Y. Goyal, A. Mohapatra, D. Parikh, and D. Batra, "Towards transparent ai systems: Interpreting visual question answering models," arXiv:1608.08974, 2016.

[37] A. W. Harley, "An interactive node-link visualization of convolutional neural networks," in *Proc. Advances Visual Comput.*, 2015, pp. 867–877.

[38] F. Hohman, N. Hodas, and D. H. Chau, "ShapeShop: Towards understanding deep learning representations via interactive experimentation," *CHI, Extended Abstracts*, pp. 1694–1699, 2017.

[39] M. Kahng, P. Andrews, A. Kalro, and D. H. Chau, "ActiVis: Visual exploration of industry-scale deep neural network models," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 88–97, Jan. 2018.

[40] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," arXiv:1506.02078, 2015.

[41] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in nlp," arXiv:1506.01066, 2015.

[42] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu, "Analyzing the training processes of deep generative models," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 77–87, Jan. 2018.

[43] Y. Ming, S. Cao, R. Zhang, Z. Li, and Y. Chen, "Understanding hidden memories of recurrent neural networks," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2017.

[44] A. P. Norton and Y. Qi, "Adversarial-Playground: A visualization suite showing how adversarial examples fool deep learning," in *Proc. IEEE Symp. Vis. Cyber Security*, 2017, pp. 1–4.

[45] C. Olah, "Visualizing MNIST," *Olah's Blog*, 2014. [Online]. Available: http://colah.github.io/posts/2014–10-Visualizing-MNIST/

[46] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, 2018.

[47] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova, "DeepEyes: Progressive visual analytics for designing deep neural networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 98–108, Jan. 2018.

[48] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea, "Visualizing the hidden activity of artificial neural networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 101–110, Jan. 2017.

[49] C. Robinson, F. Hohman, and B. Dilkina, "A deep learning approach for population estimation from satellite imagery," in *Proc. SIGSPATIAL Workshop Geospatial Humanities*, 2017, pp. 47–54.

[50] X. Rong and E. Adar, "Visual tools for debugging neural language models," in *Proc. ICML Workshop Vis Deep Learn.*, 2016.

[51] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, "Embedding Projector: Interactive visualization and interpretation of embeddings," in *Proc. Neural Inf. Process. Syst. Workshop Interpretable ML Complex Syst.*, 2016.

[52] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 667–676, Jan. 2018.

[53] J. Wang, L. Gou, H. Yang, and H.-W. Shen, "GANViz: A visual analytics approach to understand the adversarial game," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 6, pp. 1905–1917, Jun. 2018.

[54] B. Webster, "Now anyone can explore machine learning, no coding required," *Google Official Blog*, 2017. [Online]. Available: https://www.blog.google/topics/machine-learning/now-anyone-can-explore- machine-learning-no-coding-required/

[55] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Proc. ICML Deep Learn. Workshop*, 2015.

[56] T. Zahavy, N. Ben-Zrihem, and S. Mannor, "Graying the black box: Understanding DQNs," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1899–1908.

[57] H. Zeng, H. Haleem, X. Plantaz, N. Cao, and H. Qu, "CNN Comparator: Comparative analytics of convolutional neural networks," in *Proc. Workshop Visual Analytics Deep Learn.*, 2017.

[58] W. Zhong, C. Xie, Y. Zhong, Y. Wang, W. Xu, S. Cheng, and K. Mueller, "Evolutionary visual analysis of deep neural networks," in *Proc. Int. Conf. Mach. Learn. Workshop Vis Deep Learn.*, 2017.

[59] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[60] Z. C. Lipton, "The mythos of model interpretability," arXiv:1606.03490, 2016.

[61] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Process.*, vol. 73, pp. 1–15, 2018.

[62] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," arXiv:1706.07269, 2017.

[63] A. Weller, "Challenges for transparency," in *Proc. Int. Conf. Mach. Learn. Workshop Human Interpretability ML*, 2017.

[64] F. Offert, ""I know it when I see it". visualization and intuitive interpretability," in *Proc. Conf. Neural Inf. Process. Syst. Symp. Interpretable ML*, 2017.

[65] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, et al., "Google's multilingual neural machine translation system: enabling zero-shot translation," arXiv:1611.04558, 2016.

[66] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," arXiv:1702.04595, 2017.

[67] L. Li, J. Tompkin, P. Michalatos, and H. Pfister, "Hierarchical visual feature analysis for city street view datasets," in *Proc. Workshop Visual Analytics Deep Learn.*, 2017.

[68] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Univ. Montreal, Montreal, Canada, Tech. Rep. 1341, 2009, vol. 1341.

[69] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Why did you say that? visual explanations from deep networks via gradient-based localization," arXiv:1610.02391, 2016.

[70] 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 *Proc. Advances Neural Inf. Process. Syst.*, 2016.

[71] K. Patel, J. Fogarty, J. A. Landay, and B. Harrison, "Investigating statistical machine learning as a tool for software development," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2008, pp. 667–676.

[72] T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf, "Principles of explanatory debugging to personalize interactive machine learning," in *Proc. 20th Int. Conf. Intell. User Interfaces*, 2015, pp. 126–137.

[73] B. Nushi, E. Kamar, E. Horvitz, and D. Kossmann, "On human intellect and machine failures: Troubleshooting integrative machine learning systems," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1017–1025.

[74] E. Alexander and M. Gleicher, "Task-driven comparison of topic models," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 320–329, Jan. 2016.

[75] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica, "Ad click prediction: A view from the trenches," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1222–1230.

[76] M. Kahng, D. Fang, and D. H. P. Chau, "Visual exploration of machine learning results using data cube analysis," in *Proc. SIGMOD Workshop Human-In-the-Loop Data Analytics*, 2016, Art. no. 1.

[77] W. Yu, K. Yang, Y. Bai, H. Yao, and Y. Rui, "Visualizing and comparing convolutional neural networks," arXiv:1412.6631, 2014.

[78] M. Wattenberg, F. Vigas, and I. Johnson, "How to use t-SNE effectively," *Distill*, 2016.

[79] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[80] A. Nguyen, J. Yosinski, and J. Clune, "Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks," in *Proc. Int. Conf. Mach. Learn. Workshop Vis. Deep Learn.*, 2016.

[81] P. E. Rauber, A. X. Falcão, and A. C. Telea, "Visualizing time-dependent data using dynamic t-sne," in *Proc. Eurographics / IEEE VGTC Conf. Vis.*, 2016, vol. 2, pp. 73–77.

[82] S. Liu, P.-T. Bremer, J. J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, and V. Pascucci, "Visual exploration of semantic relationships in neural word embeddings," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 553–562, Jan. 2018.

[83] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOGgles: Visualizing object detection features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1–8.

[84] X. Rong, "word2vec parameter learning explained," arXiv:1411.2738, 2014.

[85] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist, "ConceptVector: Text visual analytics via interactive lexicon building using word embedding," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 361–370, Jan. 2018.

[86] D. S. Weld and G. Bansal, "Intelligible artificial intelligence," arXiv:1803.04263, 2018.

[87] C. Olah and S. Carter, "Research debt," *Distill*, 2017.

[88] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017.

[89] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 233–255, 2016.

[90] F. Grün, C. Rupprecht, N. Navab, and F. Tombari, "A taxonomy and library for visualizing learned features in convolutional neural networks," in *Int. Conf. Mach. Learn. Workshop Vis Deep Learn.*, 2016.

[91] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, and S. Dähne, "Learning how to explain neural networks: Patternnet and patternattribution," arXiv:1705.05598, 2017.

[92] H. Li, K. Mueller, and X. Chen, "Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation," arXiv:1712.08268, 2017.

[93] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," in *Proc. Int. Conf. Mach. Learn. Workshop Vis Deep Learn.*, 2017.

[94] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[95] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4829–4837.

[96] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5188–5196.

[97] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," arXiv:1412.6856, 2014.

[98] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," *Google Res. Blog*, 2015.

[99] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," arXiv:1412.6806, 2014.

[100] D. Wei, B. Zhou, A. Torrabla, and W. Freeman, "Understanding intra-class knowledge inside CNN," arXiv:1507.02379, 2015.

[101] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune, "Plug & play generative networks: Conditional iterative generation of images in latent space," arXiv:1612.00005, 2016.

[102] Y. Bengio, et al., "Learning deep architectures for ai," *Foundations Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–27, 2009.

[103] S. Carter and M. Nielsen, "Using artificial intelligence to augment human intelligence," *Distill*, 2017.

[104] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3156–3164.

[105] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2425–2433.

[106] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances Neural Inf. Process. Syst.*, 2014.

[107] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," arXiv:1701.00160, 2016.

[108] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 3, pp. 1249–1268, Mar. 2017.

[109] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 18, no. 11, pp. 2660–2673, Nov. 2017.

[110] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. SIGKDD Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[111] C.-Y. Tsai and D. D. Cox, "Characterizing visual representations within convolutional neural networks: Toward a quantitative approach," in *Proc. Int. Conf. Mach. Learn. Workshop Vis Deep Learn.*, 2016.

[112] S. Ritter, D. G. Barrett, A. Santoro, and M. M. Botvinick, "Cognitive psychology for deep neural networks: A shape bias case study," arXiv:1706.08606, 2017.

[113] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" *Comput. Vis. Image Understanding*, vol. 163, pp. 90–100, 2017.

[114] G. K. Tam, V. Kothari, and M. Chen, "An analysis of machine- and human-analytics in classification," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 71–80, Jan. 2017.

[115] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, pp. 671–769, 2016.

[116] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Sci.*, vol. 356, no. 6334, pp. 183–186, 2017.

[117] M. Wattenberg, F. Viegas, and M. Hardt, "Attacking discrimination with smarter machine learning," *Google Res. Website*, 2016. [Online]. Available: https://research.google.com/bigpicture/attacking-discrimination-in-ml/

[118] "Facets," *Google PAIR*, 2017. [Online]. Available: https://pair-code.github.io/facets/

[119] M. Hardt, E. Price, N. Srebro, et al., "Equality of opportunity in supervised learning," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3323–3331.

[120] E. Wall, L. Blaha, L. Franklin, and A. Endert, "Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2017.

[121] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," arXiv:1312.6199, 2013.

[122] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015.

[123] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.

[124] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," in *Proc. Int. Conf. Learn. Representations*, 2017.

[125] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," arXiv:1412.5068, 2014.

[126] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Security Privacy*, 2016, pp. 582–597.

[127] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Li, L. Chen, M. E. Kounavis, and D. H. Chau, "Shield: Fast, practical defense and vaccination for deep learning using jpeg compression," arXiv:1802.06816, 2018.

[128] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," arXiv:1607.02533, 2016.

Fred Hohman received the BS degree in mathematics and physics. He is working toward the PhD degree at Georgia Tech's College of Computing. His research combines HCI principles and ML techniques to improve deep learning interpretability. He won the NASA Space Technology Research Fellowship. He won SIGMOD'17 Best Demo, Honorable Mention; Microsoft AI for Earth Award for using AI to improve sustainability; and the president's fellowship for top incoming PhD students. He is a member of the IEEE.

Minsuk Kahng is working toward the PhD degree in computer science at Georgia Tech. His thesis research focuses on building visual analytics tools for exploring, interpreting, and interacting with complex machine learning models and results, by combining methods from information visualization, machine learning, and databases. He received the Google PhD fellowship and NSF graduate research fellowship. His ActiVis deep learning visualization system has been deployed on Facebook's machine learning platform. He is a member of the IEEE.

Robert Pienta received the PhD degree in computational science and engineering from Georgia Tech, in 2017. He is an industry researcher in applied machine learning and visual analytics. He was an NSF FLAMEL fellow and presidential scholar at Georgia Tech. His research interests include visual analytics, graph analytics, and machine learning. In particular, the algorithms and design techniques for interactive graph querying and exploration. He is a member of the IEEE.

Duen Horng (Polo) Chau is an associate professor with Georgia Tech. His research bridges data mining and HCI to make sense of massive datasets. His thesis won Carnegie Mellon's CS Dissertation Award, Honorable Mention. He received awards from Intel, Google, Yahoo, LexisNexis, and Symantec; He won paper awards at SIGMOD, KDD and SDM. He is an ACM IUI steering committee member, IUI15 co-chair, and IUI19 program co-chair. His research is deployed by Facebook, Symantec, and Yahoo. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.