Deep Learning: Systems and Responsibility

Abdul Wasay
Harvard University

Subarna Chatterjee
Harvard University

Stratos Idreos
Harvard University

ABSTRACT

Deep learning enables numerous applications across diverse areas. Data systems researchers are also increasingly experimenting with deep learning to enhance data systems performance. We present a tutorial on deep learning, highlighting the data systems nature of neural networks as well as research opportunities for advancements through data management techniques. We focus on three critical aspects: (1) classic design tradeoffs in neural networks which we can enrich through a systems and data management perspective, e.g., thinking critically about storage, data movement, and computation; (2) classic design problems in data systems which we can reconsider with neural networks as a viable design option, e.g., to replace or help system components that make complex decisions such as database optimizers; and (3) essential considerations for responsible application of neural networks in critical human-facing problems in society and how these also link to data management and performance considerations. While these are seemingly a diverse set of rich topics, they are strongly interconnected through data management, and their combination offers rich opportunities for future research.

ACM Reference Format:
Abdul Wasay, Subarna Chatterjee, and Stratos Idreos. 2021. Deep Learning: Systems and Responsibility. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), June 20–25, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3448016.3457541

1 INTRODUCTION

Deep learning has seen an increasing amount of success. Deep neural networks are powerful computational models that can learn intricate and complex patterns directly from data [59]. By utilizing deep neural networks, researchers across several research communities can now solve problems that had evaded them for decades. These intricate computational models have come to power countless aspects of our society today. For instance, by applying deep learning models, researchers can localize and label objects within an image 3× more accurately than the best classical methods [65]. In addition, deep neural networks can translate languages, help drive cars, and diagnose various diseases [12, 51, 71, 88]. Database systems researchers are also experimenting with neural networks and applying them to enhance database system performance [29, 54, 111].

SIGMOD '21, June 20–25, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8343-1/21/06…$15.00
https://doi.org/10.1145/3448016.3457541

This widespread impact also raises critical concerns on designing and responsibly applying deep learning. These concerns arise because of three aspects of deep learning models: (a) their performance on any task is linked to the data we feed them with, (b) their training process is stochastic, and (c) using them consumes vast amounts of computing resources and energy.

This tutorial consists of three parts that blend deep learning with concepts and considerations that have to do with data management, data systems, and responsibility.

Part 1: A Systems Perspective on Deep Learning. Deep neural networks may consist of hundreds of compute-intensive layers of neurons trained through an iterative training procedure and deployed across heterogeneous devices. The primary challenge is to design methods that yield accurate deep neural networks while minimizing training time, inference time, and memory requirements.

In the first part of the tutorial, we provide the necessary background on neural networks from a systems perspective. We present various techniques to optimize for the core metrics looking holistically at both accuracy and system-level metrics. Since, in practice optimizing for one metric requires sacrificing another, we organize the various techniques in a framework that classifies them regarding how these techniques tradeoff between two or more metrics.

Part 2: Deep Learning for Data Systems. In the last five years, an increasing amount of research applies deep learning-based solutions to address a wide array of classical data systems problems – these range from efficient query processing to data exploration and visualization frameworks.

In the second part of the tutorial, we provide an overview of these techniques across various database system components. We explain how efficient data movement and computation techniques can make it viable to apply deep learning to even more components of a database system where we need both accuracy and efficient inference (e.g., the optimizer). We categorize approaches with respect to whether deep learning is used to enhance engineering/design decisions or replace an existing system component.

Part 3: Responsible Deep Learning. When deep learning is applied to user-facing applications, then there are new challenges to consider. For example, when we use deep learning to make decisions that can directly impact humans, it is important to understand how deep learning models made these decisions and why. In the third part of the tutorial, we explore responsibility across three dimensions: fairness, interpretability, and environmental impact. Again, we bring attention to challenges and opportunities for future research that are amenable to data management and systems solutions such as data provenance and efficient processing.

Audience. The tutorial is designed for an audience with a basic data management background (students, academics, researchers,

and industry practitioners) and does not assume any background in neural networks or machine learning.

Website. We make available the tutorial text with linked citations and additional material, including slides, video, and a references navigator at: daslab.seas.harvard.edu/dl-sys-responsibility/ .

Output. The learning outcomes are as follows:

(1) understanding deep learning from a systems perspective.
(2) understanding the tradeoffs between accuracy, training time, inference time, and memory usage.
(3) exposure to systems-oriented research that exploits as well as navigates these tradeoffs.
(4) exposure to machine learning in systems research that utilizes neural networks to improve data system components.
(5) an appreciation for the societal issues regarding fairness, interpretability, and environmental impact that applying deep learning to the world around us brings up.
(6) exposure to new opportunities for data management and systems researchers across different contexts: data sharing, memory management, storage designs, distributed systems, and **how a holistic understanding of algorithm design, systems performance, and responsibility is necessary**.

2 PART 1: A SYSTEMS PERSPECTIVE ON DEEP LEARNING

Fundamentals of Deep Neural Networks. Deep neural networks today have dozens of layers consisting of simple but non-linear modules. Every layer in a neural network progressively transforms its input from one level of representation to a more abstract level, which better captures aspects of the data set that are meaningful for a classification or detection task. Once a deep neural network architecture has been specified, the training process is composed of a set of alternating forward and backward passes until a specified metric (usually training accuracy) converges.

We draw examples from convolutional neural networks in this tutorial. This class of networks, first introduced for computer vision tasks, is utilized in diverse applications such as drug discovery, machine translation, and query optimization [102]. However, the principles we discuss, such as the tradeoff between accuracy and resources, apply to a wide range of deep learning paradigms.

A Query Processing Analogy. We can draw an analogy here between a query processing pipeline and a deep neural network. Like operators in a query processing pipeline, layers in neural networks function as semantic filters, only letting relevant information (i.e., patterns) go through to the next layer [59, 114]. Every layer has both logic and weights associated with it. Training sets up the pipeline (i.e., tunes the weights), and during deployment, we pass every data item through this predefined pipeline.

Computation and Data Movement. Training and deploying deep neural networks trigger a large amount of computation on huge data sets [22]. The large data sets needed to train neural networks, the parameters at every layer, and the intermediate results all contribute to this data movement and computation. For instance, Wide ResNets, a state-of-the-art class of neural network architectures, can have up to a million weights per layer and over 40 layers in a single network [113].

Metrics. There are two categories of metrics: quality-related metrics and resource-related metrics. Quality-related metrics include training accuracy, generalization accuracy, and robustness, and they quantify how good a deep neural network is at performing the task it is trained for. On the other hand, resource-related metrics such as training time, inference speed, and memory usage quantify how efficiently a network can achieve the desired result [22].

Systems Tradeoffs in Deep Learning. Deep learning metrics are tightly connected. The relationships between various metrics, though, are non-linear and depend on the network architecture, the training process, and the hardware in ways that we cannot consistently map out. A challenge for researchers and practitioners is to understand these relationships and design techniques that can strike specific tradeoffs between two or more of these metrics [106].

We classify existing techniques that improve deep learning efficiency in terms of how they tradeoff between various metrics. In particular, we look at three such classes of tradeoffs: (i) techniques to reduce the training and inference cost (both in terms of computation and memory overhead) at the expense of a possible decrease in accuracy, (ii) methods to reduce the training and inference cost by dedicating setup time to optimize deep learning pipelines before training or deploying them, and (iii) research that trades off training and inference time to reduce memory usage.

2.1 Accuracy vs. Time Efficiency

Compressing Deep Neural Networks. Compression reduces the overall footprint of a neural network and is a prevalent technique to reduce the training time, memory usage, and inference time. The accuracy of the compressed network might be affected depending on whether the technique is lossy or lossless. Compression techniques in deep learning fall under three categories: (i) Quantization, (ii) Parameter pruning, and (iii) Knowledge distillation [10].

Quantization approaches reduce the size of neural networks by decreasing the precision of network parameters and intermediate results. Both scalar and vector quantization techniques have been explored that replace the original data by a set of quantization codes and a codebook. This codebook may be constructed in a lossless manner (e.g., Huffman encoding) or in a lossy manner (e.g., low-bit fixed-points or K-means); How lossy this codebook is determines whether or not the quantization affects accuracy. There are proposals to quantize weights, intermediate results, or both down to various precision levels, ranging from eight bits to just one bit, e.g., in the case of Binary Neural Networks [17, 21, 64]. Similarly, there are proposals to replace floating point numbers in networks with integers that are more efficient to operate on [44, 110]. Finally, some approaches dynamically learn how to quantize based on the given architecture, data set, and hardware [11, 46, 81, 116].

Neural network pruning approaches operate under the premise that many parameters are either unnecessary or not extremely useful and design techniques to remove those parameters [4]. Various approaches prune at different granularity, e.g., parameter-level [27], filter-level [31, 62], and network-level [58]. Similarly, various approaches can be used to inform what to prune. These range

from magnitude-based approaches (i.e., prune parameters with low-magnitude) [27] to loss-based approaches (i.e., prune parameters that have less effect on a given loss function) [65] to approaches that automatically learn which parameters to prune [6].

Knowledge distillation approaches can reduce the memory and computational footprint of deep neural networks by transferring the function that is learned by large networks into smaller networks [34]. This class of methods has been used to improve inference at the edge [79], to accelerate ensemble training [34], and to bootstrap the training of large networks [117].

Training and Deploying Deep Ensembles. This tradeoff between accuracy and various resources has also been extensively explored while training ensembles of deep neural network, where not just one but multiple networks are trained to perform the same task [107]. Various approaches such as SnapShot Ensembles and Fast Geometric Ensembles generate ensembles of deep neural networks by training a single neural network model once and saving copies of the model at various points in the training trajectory [18, 36]. Additionally, approaches like TreeNets and MotherNets capture the structural similarity between different networks in a heterogenous ensemble and train for it just once [60, 106]. All these approaches provide lower accuracy than the baseline approach of training every member of the ensemble from scratch but require significantly less training time. MotherNets and TreeNets also reduce the memory usage and inference time.

Relaxing Constraints in Distributed Settings. In distributed settings, the major contributor to both the inference and the training cost is the overhead of communicating between different nodes. Recent work reduces the communication cost by progressively relaxing the constraint that all nodes need to have a fresh copy of the network parameters at all times. Local SGD, for instance, trains various copies of the network in parallel and averages their parameters after a configurable number of training rounds [91]. Tangential to this are those methods that reduce the communication cost by compressing gradients that get communicated between devices [63]. Recently, there has also been work in prioritizing what to communicate between machines [47].

2.2 Optimization Time vs. Training and Inference Time Efficiency

Optimizing for Distributed Training. FlexFlow automatically figures out a parallelization strategy given the network architecture and the hardware setup. In particular, FlexFlow introduces an additional optimization step that uses simulation and guided search to decide on a near-optimal parallelization strategy [48, 67]. This optimize-then-parallelize framework has been extended by various subsequent systems to include memory constraints on devices as well as consider devices with heterogenous compute power and communication links [2, 52].

Optimizing for Inference. Additionally, some approaches use an optimization step to tailor a model for inference. MorphNet, for instance, iteratively resizes a network based on resource constraints that can be in terms of either model size or computational requirements [23]. Mnasnet and Netadapt further extend this approach to take into account particulars of the edge device that the network is to be deployed on [94].

2.3 Training Time vs. Memory Efficiency

Recomputing Intermediate Results. Many approaches utilize the observation that intermediate results (produced during the forward pass) don't need to be stored but can be recomputed when needed. This reduces the overall memory that is needed to train a neural network at the cost of some extra training time. In particular, there are methods that store equidistant layers (i.e., checkpoints) in the network and can train a network with geometrically less memory at the cost of an additional forward pass [9, 25]. Recently, Checkmate generalizes this framework and can find an optimal checkpointing strategy given any amount of memory [45] .

Offloading Intermediate Results. Another set of solutions to reduce the memory overhead of deep neural networks is to offload the intermediate results to CPU memory [76]. This again results in some additional training time overhead as the results need to be reread from a slower memory subsystem.

Data Management Opportunities. We will present opportunities to apply established techniques from database management, such as vectorized processing, end-to-end optimization, and layout design, to further improve these classes of tradeoff in deep learning. We will also highlight how paradigms from the data management world, such as standardized benchmarks, query languages, and declarative interfaces can further extend the usability of deep learning models.

3 PART 2: DEEP LEARNING IN DATA SYSTEMS

We now discuss research that designs deep learning-based methods to enhance, automate, and even replace various data systems' decision-making components.

Improving Query Optimization. Deep neural networks are deployed at various stages of the query optimization pipeline. They are used to improve selectivity estimates for queries targeting multiple attributes [29, 30]. There are proposals to use deep neural networks to generate query plans directly [102]. Finally, deep reinforcement learning techniques are used to tune various design knobs within a database system, such as the data layout and memory allocation to various components [49, 61, 115].

Enhancing Access Methods. Neural networks are also used to replace or enhance access methods and index structures such as B-trees and Bloom filters. Learned indexes, for instance, can take the form of deep neural networks and learn the mapping between data items and their location, [54]. For instance, SageDB is a database system that proposes a holistic database system designed around learned components [53] and MLWeaving is an in-memory data structure that enables faster learning of low-precision data [104].

Our own work on self-designing data systems [37] and a Calculus of Data structures [38–40] show how to use neural networks to navigate massive complex design spaces of fine-grained system designs and learn cost models on how these primitives behave without having to code the target system designs.

Enabling Data Exploration. Data exploration is an area of active research within the data systems community to design tools and techniques to enable a data scientist to understand the various properties of new data sets [105, 108]. Deep reinforcement learning techniques are applied to learn from user interactions and automatically guide them to insights in their data sets [66, 82, 99]. Recurrent neural networks are also used to enable natural language querying of databases and generate exploratory queries [3, 87]. Lastly, techniques inspired by deep word embeddings are used to enhance similarity search within relational databases [15].

Compressing and Integrating Data. Finally, neural networks are also used to compress relational data sets and enhance data integration through more accurate entity matching [41, 70]. For instance, Bit-Swap uses hierarchical latent variable models to outperform benchmark compressors.

Research Opportunities. We will present opportunities to rethink several decision-making components within database systems and extend them using deep learning models, including low-level and high-level design decisions from data structure design to query scheduling. Then, we will discuss opportunities to exploit the representational capability of deep data embeddings to learn semantic information about the data set that can inform both query processing and guide database users. Finally, there are open questions on extending, scaling, and managing deep learning-based access methods and data models.

4 PART 3: RESPONSIBLE DEEP LEARNING

4.1 Fairness

The quality of deep learning models is measured in terms of accuracy metrics (e.g., training error and test accuracy) and systems metrics (e.g., training time and inference time). While these metrics capture how efficiently and accurately deep learning models estimate specific patterns in the data, they do not give us a way to understand and analyze potential bias present in the decisions that such models make [13, 92].

The inherent technical problem is that the quality of the results (predictions) we get from deep learning models depends on the specific data set used for training the model. This is true for both the data chosen for training and the labels used to describe the training data. Both of these are choices made by humans who carry and transfer their biases to the data. Similarly, deep learning model building and design come with several tuning choices and judgment calls to stop training and model tuning. Again these are all human choices that carry the biases of human designers.

The absence of social biases awareness within the design process and deep learning systems can have several negative implications starting from the marginalization of vulnerable groups to the exacerbation of prejudice and discrimination. These effects are especially problematic for applications that deal with social data, e.g., face recognition or mortgage decisions. It is essential to ensure equitable predictions across all groups, and thus it is crucial to integrate *fairness* as an evaluation metric for deep learning model design.
In this section of the tutorial, we first review different attempts to formalize fairness and then describe various techniques to enhance fairness in data and in algorithms.

Formalizing Fairness. Multiple frameworks have been introduced to formalize the notion of fairness and its implications on machine learning data, methods, and results [24, 69, 89]. A general framing of fairness centers around three questions within the application context: (i) Is it fair to apply machine learning to a problem? (ii) If so, is there a fair way to do so? and (iii) Even if there is a fair method, then are the results produced fair? [89]. A system can be fair only when it provides contextual answers to these questions and allows those affected by it to challenge or confirm fairness.

Recent studies utilize frameworks from areas outside Computer Science to conceptualize fairness. One such set of studies borrows from critical race theory and advocate understanding the instability and multi-dimensional nature of various demographic categories (such as race and gender). They use this notion of instability to inform both the design and evaluation of algorithms [28, 35]. Similarly, economic models of fairness such as equality of opportunity [32], and transparent and accountable models for sensitive applications such as criminal sentencing and credit scoring [55] are popular examples. Most relevant to the data systems community is a recent study that advocates for developing a shared definition of fairness across the board in engineering and data teams [73].

Fairness in Data. Deep learning pipelines heavily rely on training data sets, which can replicate various pre-existing social biases and inequality such as ethnicity- or gender-based discrimination. A recent report highlights the existence of a significantly large number of misleading gender stereotypes within data sets at Google and how that carries over to resulting word embeddings [72].

There is a growing body of research to address the question of fairness in data. First, there are approaches to ensure fairness at the data collection level advocating for consent, power, inclusivity, and transparency [20, 68]. For data sets that have already been collected, there are proposals to accompany them with metadata explaining their composition and collection process so that users can use them in an informed manner [19, 93]. Complementary to this are data pre-processing techniques to filter through already collected data to generate a training set that is less biased and more diverse [8, 85]. Finally, there is work on augmenting and generating data to have better privacy and fairness guarantees [74, 78].

Fairness in Algorithm. Unfairness can also occur at the algorithmic level, i.e., during the design and training of the deep neural network. In one such example of algorithmic unfairness, a deep neural network model could infer the gender of a person from images of their retina even though gender was not included in the training data set [14, 109].

There are efforts to mitigate unfairness at the algorithmic level both during and after training. A popular set of techniques is to use adversarial learning, where two models are trained – the predictor model that learns the most informative representation possible from data and an adversarial model that reduces the predictor's capability to learn about protected attributes [16]. Some methods can remove bias from an already trained deep neural network. These methods proceed by detecting and removing neurons or parameters from the network strongly correlated with protected attributes [50].

Data Management Opportunities. Both aspects of unfairness mentioned above deal with concepts and properties familiar to the

data management and data systems community. Properly managing, ensuring, and propagating data properties through ontologies and systems optimizations to allow for more complex models or more models (and thus better accuracy) are critical directions that can help achieve a positive push for more ethical deep learning.

4.2 Interpretability

The results given by deep learning models used in practice are extremely hard to understand and reason about. This is because they have been generated by networks with millions of parameters trained through a stochastic process. Theory lags far behind practice: Robust theoretical analysis exists only for very simple models trained on very small data sets [1]. Interpretable deep learning has emerged as a sub-field of deep learning that seeks to augment the design, training, and deployment of deep learning models to make them understandable to humans [7, 75, 84]. For instance, when applying deep learning to decide whether to provide loans to an individual, it is essential not only to have a final decision but also to list reasons on which such a decision was made (so that it can be verified). In addition, when applying deep learning in health care it is critical to know exactly why certain suggestions are made by the models as any wrong decision can be catastrophic. Overall, interpretable deep learning enables experts and non-experts to understand, verify, and trust decisions made by neural networks.

We provide an overview of existing work on interpretable deep learning across three directions: dimensionality reduction, visualization, and model surrogacy.

Dimensionality Reduction. Deep learning pipelines are replete with extremely high-dimensional data such as training data sets and evolving neural network parameters. Dimensionality reduction techniques enable understanding high-dimensional data by converting it into a low-dimensional representation while preserving meaningful properties [80, 96, 100]. A widely-used algorithm for dimensionality reduction in deep learning pipelines is called t-distributed Stochastic Neighbor Embedding (t-SNE) that preserves local similarities present in high-dimensional data sets [100]. For instance, t-SNE can convert the MNIST training data set (with 784-dimensional images) into a two-dimensional representation while maintaining clusters present in the data set, i.e., images belonging to different classes in high-dimensional data stay in different clusters in the low-dimensional representation. T-SNE and its variants, such as Isomap and Locally Linear Embedding, can also be applied to parameters of deep neural networks and its outputs. We can use the resulting low-dimensional representation for debugging, exploration, and visualization, making it much easier to understand the data and potential biases being present.

Visualization of Relationships. Various methods help visualize different aspects of deep learning to understand the trilateral relationship between input data items, parameters of a deep learning model, and the outputs it produces [75]. For example, such visualization can be very useful when a data scientist is interested in mapping out sub-parts of a deep neural network responsible for recognizing certain features present in an input image. Activation Maximization is one such widely-used technique to achieve this. Activation Maximization synthesizes an input that maximally activates a specific part of the neural network [114]. This synthetic input indicates the features that a specific part of a neural network recognizes. Another set of techniques called DeconNet takes a specific layer in a convolutional neural network and operates in the reverse direction to figure out patterns in an input image responsible for the activation produced by that layer [97]. This is often used for debugging the scenario when the network produces incorrect outputs. Finally, there is the technique of Network Inversion that takes only the local information present at a layer in a neural network and reconstructs the input [83]. This visualizes what aspects of an input (e.g., image) are preserved at every layer. We can apply all these techniques and their variants at various resolutions of a neural network, ranging from a neuron to a layer or even a set of layers. Together, they can construct detailed visualizations of how inputs to the deep neural network get converted to decisions. Such approaches do not solve the problem of bias in the data automatically, but can play a drastic role in helping human designers more easily spot bias in the data or design and act to fix it.

Model Surrogacy. Finally, a very standard approach used in practice is to approximate a deep neural network's decision function with self-explanatory surrogate models [84]. These surrogate models can be models which are straightforward or easier to interpret such as linear classifiers, mixtures of decision trees, or even less complex neural network models. One popular approach is Local Interpretable Model-Agnostic Explanations (LIME) [77]. Given an input and a deep neural network, LIME produces a linear surrogate model that explains the contribution made by all input features to the decision made by the deep neural network. LIME proceeds by first defining a probability distribution around the input data point and, then, learning a linear model that best matches the output produced by the neural network on that distribution. Another approach is to use Knowledge Distillation. Here, the surrogate model takes the form of a less complex deep neural network model that mimics the deep neural network's decision function. Overall, we can combine model surrogacy approaches to produce explanations at different semantic levels. For instance, this could be at the level of pixels, image features, or classes of images.

Frameworks and Systems. Methods to interpret deep learning models have been implemented both as a part of existing deep learning frameworks and standalone packages in various programming languages. Tensorboard is a neural network visualization and debugging framework integrated with TensorFlow. Tensorboard has tools for visualization of an end-to-end deep learning pipeline and can provide visual summaries of the training data, the training process, and the trained deep neural network. TorchRay and Captum provide implementations of various interpretable deep learning algorithms in PyTorch. Other examples of such tools include DeepExplain and iNNvestigate that support different methods to visualize, debug, and answer what-if questions.

In addition to these frameworks, there are proposals for full systems for efficient visualization and debugging of trained deep learning models. DeepVis is a system to visualize activations in deep neural networks as they train [112]. Mistique is a system to efficiently store, manage, and query deep learning models [101].

Deepbase provides a declarative interface to specify and test hypotheses and what-if queries on trained models [86].

Data Management Opportunities. Various techniques related to interpretable deep learning, such as dimensionality reduction and data visualization, have been extensively explored in database research for understanding relational data. Many optimizations, such as smart caching and aggregations, explored in the data management context can also be explored to improve deep learning interpretability at scale. In addition to this, there are opportunities to design end-to-end deep learning systems with in-built data and model tracking during design, training, and deployment phases. Here, various ideas explored in provenance-aware systems can be applied to build interpretable deep learning systems.

4.3 Environmental Impact

Deep learning pipelines require an increasing amount of energy to design, train, and deploy. Computational resources needed to produce state-of-the-art deep learning models double every three months and have grown by over 300000× from 2015 to 2020 [98]. For instance, even a single training phase of a large deep learning model can emit as much CO_2 as five cars produce throughout their lifetimes. This environmental impact is set to grow exponentially. This is because of: (i) growing training data sets and model sizes as applications that need to employ deep learning get more complex, (ii) lengthy feature generation, model design, and tuning steps where designers have to train a model numerous times, and (iii) increasing pressure for efficient deployment of models that can produce results in the order of milliseconds.

Carbon Footprints. As a first step, it is critical to be able to capture the energy efficiency of deep learning models and use that as a metric in model design. For instance, Machine Learning Emissions Calculator and the Green Algorithms Project can provide a detailed breakdown of a model's carbon footprint based on hardware, cloud provider, and region [56, 57].

Green Hardware and Cloud Providers. Next, there are opportunities to evaluate hardware and cloud providers. One such method is to track the Power Usage Effectiveness Ratio (PUE) of cloud providers and FLOPs/W of hardware and make choices that maximize these metrics given a workload [43, 56, 95]. Additionally, there is growing research to investigate new paradigms such as photonics and quantum computing to design specialized hardware that drastically improves the metric of FLOPs/Watt [26].

Resolution-Setting Frameworks. Last, but not least, setting resolutions and tracking progress is a critical aspect. For instance, Microsoft plans to be carbon-negative by 2030 [90] and Apple and Amazon plan to attain carbon-neutrality by 2030 and 2050, respectively [5, 33] by planning to utilize diverse renewable power sources such as wind, solar, advanced nuclear, enhanced geothermal, and green hydrogen for their data centers [42, 103].

Data Management Opportunities. Due to the origins of the issues being the large data sizes and computational costs, data management and systems research can play a drastic role here. We outline ongoing and open directions that data systems researchers and practitioners can take to reduce deep learning's environmental impact. First, there are opportunities to rethink model design, training, and deployment to utilize existing hardware better, e.g., utilizing the massive mismatch between compute and IO capacities of modern GPUs to design models that perform more compute per IO or allocating deep learning jobs in the cloud to minimize energy waste. We can also build deep learning systems that enable reuse and caching across all stages, including data sourcing, design, training, and deployment.

5 PRESENTERS

Abdul Wasay (awasay@seas.harvard.edu) is a Ph.D. candidate in Computer Science working with prof. Stratos Idreos and the Data Systems Lab at Harvard University. His research is at the intersection of systems and machine learning. Wasay identifies co-design opportunities between the two fields to develop techniques that accelerate data science and deep learning pipelines by removing computation and data movement bottlenecks. His work appears at leading systems and machine learning venues such as SIGCOMM, SIGMOD, MLSys, and ICLR. In 2017, Wasay received the 'most reproducible paper award' at SIGMOD. He earned the 2019 Archer-Cornfield Fellowship and spent a year as a visiting faculty member at Ashesi University. Before joining Harvard, Wasay was an undergraduate at Lahore University of Management Sciences (LUMS). He also spent a summer at EPFL and another at HP Research.

Subarna Chatterjee (subarna@seas.harvard.edu) is a post-doc at Harvard University advised by Stratos Idreos. She is also a core member of the Embedded Ethics group at Harvard University. Her research is about data systems design on the cloud using learning approaches to optimize system design. In 2016, she was selected as one of the "10 Women in Networking/Communications That You Should Watch" and one of the young scientists to attend the Heidelberg Laureate Forum.

Stratos Idreos (stratos@seas.harvard.edu) is an associate professor of Computer Science at Harvard University where he leads the Data Systems Laboratory. His research focuses on making it easy and even automatic to design workload and hardware conscious data structures and data systems with applications on relational, NoSQL, data science, machine learning and data exploration problems. Stratos was awarded the ACM SIGMOD Jim Gray Doctoral Dissertation award for his thesis on adaptive indexing as well as the 2011 ERCIM Cor Baayen award from the European Research Council on Informatics and Mathematics. He won the 2011 Challenges and Visions best paper award in the Very Large Databases conference as well as 'best of conference' selections at VLDB 2012 and SIGMOD 2017. In 2015 he was awarded the IEEE TCDE Rising Star Award from the IEEE Technical Committee on Data Engineering for his work on adaptive data systems. Stratos is also a recipient of the IBM zEnterpise System Recognition Award, a Facebook Faculty award, a NetApp Faculty award, an NSF Career award, and a Department of Energy Early Career award.

Acknowledegments. This work is partly funded by the USA Department of Energy Project DE-SC0020200.

REFERENCES

[1] Md Zahangir Alom, Tarek M Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C Van Essen, Abdul AS Awwal, and Vijayan K Asari. 2019. A state-of-the-art survey on deep learning theory and architectures. *Electronics* 8, 3 (2019), 292.

[2] Neil Band. 2020. MemFlow: Memory-Aware Distributed Deep Learning. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA.

[3] Ori Bar El, Tova Milo, and Amit Somech. 2020. Automatically Generating Data Exploration Sessions Using Deep Reinforcement Learning. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1527–1537. https://doi.org/10.1145/3318464.3389779

[4] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the State of Neural Network Pruning?. In *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze (Eds.), Vol. 2. 129–146.

[5] Justine Calma. 2019. Jeff Bezos pledges that Amazon will swiftly combat climate change. (2019). https://www.theverge.com/2019/9/19/20873834/amazon-sustainability-jeff-bezos-climate-change-pledge-emissions-paris-accord

[6] Miguel A Carreira-Perpinán and Yerlan Idelbayev. 2018. "Learning-Compression" Algorithms for Neural Net Pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8532–8541.

[7] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (2019), 832.

[8] L Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. 2016. How to be fair and diverse? *arXiv preprint arXiv:1610.07183* (2016).

[9] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training Deep Nets with Sublinear Memory Cost. *CoRR* abs/1604.06174 (2016).

[10] Jian Cheng, Pei-song Wang, Gang Li, Qing-hao Hu, and Han-qing Lu. 2018. Recent advances in efficient computation of deep convolutional neural networks. *Frontiers of Information Technology & Electronic Engineering* 19, 1 (2018), 64–77.

[11] Jungwook Choi, Swagath Venkataramani, Vijayalakshmi (Viji) Srinivasan, Kailash Gopalakrishnan, Zhuo Wang, and Pierce Chuang. 2019. Accurate and Efficient 2-bit Quantized Neural Networks. In *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. 348–359.

[12] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. 2015. Compression Artifacts Reduction by a Deep Convolutional Network. In *IEEE International Conference on Computer Vision*.

[13] Marina Drosou, HV Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. 2017. Diversity in big data: A review. *Big data* 5, 2 (2017), 73–84.

[14] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2020. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems* (2020).

[15] Karima Echihabi. 2020. High-Dimensional Vector Similarity Search: From Time Series to Deep Network Embeddings. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 2829–2832. https://doi.org/10.1145/3318464.3384402

[16] Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. *arXiv preprint arXiv:1808.06640* (2018).

[17] Joshua Fromm, Meghan Cowan, Matthai Philipose, Luis Ceze, and Shwetak Patel. 2020. Riptide: Fast End-to-End Binarized Neural Networks. In *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze (Eds.), Vol. 2. 379–389.

[18] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *Advances in Neural Information Processing Systems*.

[19] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010* (2018).

[20] R. Stuart Geiger, Kevin Yu, Yanlai Yang, Mindy Dai, Jie Qiu, Rebekah Tang, and Jenny Huang. 2020. Garbage in, Garbage out? Do Machine Learning Application Papers in Social Computing Report Where Human-Labeled Training Data Comes From? *(FAT* '20)*.

[21] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115* (2014).

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

[23] Ariel Gordon, Elad Eban, Ofir Nachum, Bo Chen, Hao Wu, Tien-Ju Yang, and Edward Choi. 2018. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1586–1595.

[24] Ben Green and Salomé Viljoen. 2020. Algorithmic Realism: Expanding the Boundaries of Algorithmic Thought *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA.

[25] Andreas Griewank and Andrea Walther. 2000. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)* 26, 1 (2000), 19–45.

[26] Ryan Hamerly, Liane Bernstein, Alexander Sludds, Marin Soljačić, and Dirk Englund. 2019. Large-scale optical neural networks based on photoelectric multiplication. *Physical Review X* 9, 2 (2019), 021032.

[27] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*. 1135–1143.

[28] Alex Hanna, Emily Denton, Andrew Smart, and Jamila Smith-Loud. 2020. Towards a Critical Race Methodology in Algorithmic Fairness. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20)*. Association for Computing Machinery, New York, NY, USA.

[29] Shohedul Hasan, Saravanan Thirumuruganathan, Jees Augustine, Nick Koudas, and Gautam Das. 2020. Deep Learning Models for Selectivity Estimation of Multi-Attribute Queries. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1035–1050. https://doi.org/10.1145/3318464.3389741

[30] Rojeh Hayek and O. Shmueli. 2020. Improved Cardinality Estimation by Learning Queries Containment Rates. In *EDBT*.

[31] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-Cnn. In *IEEE international conference on computer vision*.

[32] Hoda Heidari, Michele Loi, Krishna P. Gummadi, and Andreas Krause. 2019. A Moral Framework for Understanding Fair ML through Economic Models of Equality of Opportunity *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA.

[33] Alex Hern. 2020. Facebook and Google announce plans to become carbon neutral. (2020). https://www.theguardian.com/environment/2020/sep/15/facebook-and-google-announce-plans-become-carbon-neutral

[34] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*.

[35] Lily Hu and Issa Kohler-Hausmann. 2020. What's Sex Got to Do with Machine Learning?. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA.

[36] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. 2017. Snapshot Ensembles: Train 1, Get M for Free. *5th International Conference on Learning Representations (ICLR)* (2017).

[37] Stratos Idreos, Niv Dayan, Wilson Qin, Mali Akmanalp, Sophie Hilgard, A. Ross, J. Lennon, V. Jain, Harshita Gupta, D. Li, and Zichen Zhu. 2019. Design Continuums and the Path Toward Self-Designing Key-Value Stores that Know and Learn. In *CIDR*.

[38] Stratos Idreos, Konstantinos Zoumpatianos, Manos Athanassoulis, Niv Dayan, Brian Hentschel, Michael S. Kester, Demi Guo, Lukas Maas, Wilson Qin, Abdul Wasay, and Yiyou Sun. 2018. The Periodic Table of Data Structures. *IEEE Computer Society Technical Committee on Data Engineering* 41, 3 (2018).

[39] Stratos Idreos, Kostas Zoumpatianos, Subarna Chatterjee, Wilson Qin, Abdul Wasay, Brian Hentschel, Mike Kester, Niv Dayan, Demi Guo, Minseo Kang, et al. 2019. Learning data structure alchemy. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 42, 2 (2019).

[40] Stratos Idreos, Kostas Zoumpatianos, Brian Hentschel, Michael S. Kester, and Demi Guo. 2018. The Data Calculator: Data Structure Design and Cost Synthesis from First Principles and Learned Cost Models. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 535–550. https://doi.org/10.1145/3183713.3199671

[41] Amir Ilkhechi, Andrew Crotty, Alex Galakatos, Yicong Mao, Grace Fan, Xiran Shi, and Ugur Cetintemel. 2020. DeepSqueeze: Deep Semantic Compression for Tabular Data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1733–1746. https://doi.org/10.1145/3318464.3389734

[42] Google Inc. 2020. 24/7 by 2030: Realizing a Carbon-free Future. (2020). https://www.gstatic.com/gumdrop/sustainability/247-carbon-free-energy.pdf

[43] Google Inc. 2020. Helping businesses save energy in the cloud. (2020). https://www.google.com/about/datacenters/efficiency/

[44] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2704–2713.

[45] Paras Jain, Ajay Jain, Aniruddha Nrusimha, Amir Gholami, Pieter Abbeel, Joseph Gonzalez, Kurt Keutzer, and Ion Stoica. 2020. Checkmate: Breaking the Memory Wall with Optimal Tensor Rematerialization. In *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze (Eds.), Vol. 2. 497–511.

[46] Sambhav Jain, Albert Gural, Michael Wu, and Chris Dick. 2020. Trained Quantization Thresholds for Accurate and Efficient Fixed-Point Inference of Deep Neural Networks. In *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze (Eds.), Vol. 2. 112–128. https://proceedings.mlsys.org/paper/2020/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf

[47] Anand Jayarajan, Jinliang Wei, Garth Gibson, Alexandra Fedorova, and Gennady Pekhimenko. 2019. Priority-based Parameter Propagation for Distributed DNN Training. In *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. 132–145. https://proceedings.mlsys.org/paper/2019/file/d09bf41544a3365a46c9077ebb5e35c3-Paper.pdf

[48] Zhihao Jia, Matei Zaharia, and Alex Aiken. 2018. Beyond data and model parallelism for deep neural networks. *arXiv preprint arXiv:1807.05358* (2018).

[49] Kaan Kara, Ken Eguro, Ce Zhang, and Gustavo Alonso. 2018. ColumnML: Column-Store Machine Learning with on-the-Fly Data Transformation. *Proc. VLDB Endow.* 12, 4 (Dec. 2018), 348–361. https://doi.org/10.14778/3297753.3297756

[50] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*. PMLR, 2668–2677.

[51] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *Empirical Methods in Natural Language Processing* (2014).

[52] Peter Kraft, Daniel Kang, Deepak Narayanan, Shoumik Palkar, Peter Bailis, and Matei Zaharia. 2020. Willump: A Statistically-Aware End-to-end Optimizer for Machine Learning Inference. In *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze (Eds.), Vol. 2. 147–159. https://proceedings.mlsys.org/paper/2020/file/fbd7939d674997cdb4692d34de8633c4-Paper.pdf

[53] Tim Kraska, Mohammad Alizadeh, Alex Beutel, H Chi, Ani Kristo, Guillaume Leclerc, Samuel Madden, Hongzi Mao, and Vikram Nathan. 2019. Sagedb: A learned database system. In *CIDR*.

[54] Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, and Neoklis Polyzotis. 2017. The Case for Learned Index Structures. *International Conference on Management of Data (SIGMOD)* (2017).

[55] Joshua A Kroll, Solon Barocas, Edward W Felten, Joel R Reidenberg, David G Robinson, and Harlan Yu. 2016. Accountable algorithms. *U. Pa. L. Rev.* 165 (2016), 633.

[56] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Machine Learning Emissions Calculator. (2019). https://mlco2.github.io/impact/#compute

[57] Loic Lannelongue, Jason Grealey, and Michael Inouye. 2020. Green Algorithms: Quantifying the carbon footprint of computation . (2020).

[58] Aleksandar Lazarevic and Zoran Obradovic. 2001. Effective pruning of neural network classifier ensembles. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, Vol. 2. IEEE, 796–801.

[59] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015).

[60] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David J. Crandall, and Dhruv Batra. 2015. Why M Heads are Better than One: Training a Diverse Ensemble of Deep Networks. *CoRR* abs/1511.06314 (2015).

[61] Guoliang Li, Xuanhe Zhou, Shifu Li, and Bo Gao. 2019. QTune: A Query-Aware Database Tuning System with Deep Reinforcement Learning. *Proc. VLDB Endow.* 12, 12 (Aug. 2019), 2118–2130. https://doi.org/10.14778/3352063.3352129

[62] Zhizhong Li and Derek Hoiem. 2016. Learning Without Forgetting. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*.

[63] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. 2018. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkhQHMW0W

[64] Mieszko Lis, Maximilian Golub, and Guy Lemieux. 2019. Full Deep Neural Network Training On A Pruned Weight Budget. In *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. 252–263.

[65] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. 2020. Deep learning for generic object detection: A survey. *International journal of computer vision* 128, 2 (2020), 261–318.

[66] Xinyuan Lu. 2019. Learning to Generate Questions with Adaptive Copying Neural Networks *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 1838–1840. https://doi.org/10.1145/3299869.3300100

[67] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. 2017. The Expressive Power of Neural Networks: A View from the Width. In *Advances in Neural Information Processing Systems*.

[68] Vidushi Marda and Shivangi Narayan. 2020. Data in New Delhi's Predictive Policing System *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA.

[69] Hussein Mouzannar, Mesrob I. Ohannessian, and Nathan Srebro. 2019. From Fair Decision Making To Social Equality *(FAT* '19)*. New York, NY, USA.

[70] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 19–34. https://doi.org/10.1145/3183713.3196926

[71] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning Convolutional Neural Networks for Graphs. In *International conference on machine learning*.

[72] Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. 2020. Bias in Word Embeddings *(FAT* '20)*. New York, NY, USA.

[73] Samir Passi and Solon Barocas. 2019. Problem Formulation and Fairness *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA.

[74] Haoyue Ping, Julia Stoyanovich, and Bill Howe. 2017. DataSynthesizer: Privacy-Preserving Synthetic Datasets *(SSDBM '17)*. New York, NY, USA.

[75] Zhuwei Qin, Fuxun Yu, Chenchen Liu, and Xiang Chen. 2018. How convolutional neural network see the world-A survey of convolutional neural network visualization methods. *arXiv preprint arXiv:1804.11191* (2018).

[76] Minsoo Rhu, Natalia Gimelshein, Jason Clemons, Arslan Zulfiqar, and Stephen W Keckler. 2016. vDNN: Virtualized deep neural networks for scalable, memory-efficient neural network design. In *International Symposium on Microarchitecture*.

[77] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[78] Luke Rodriguez, Babak Salimi, Haoyue Ping, Julia Stoyanovich, and Bill Howe. 2018. MobilityMirror: Bias-adjusted transportation datasets. In *Workshop on Big Social Data and Urban Computing*. Springer, 18–39.

[79] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In *International Conference on Learning Representations (ICLR)*.

[80] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[81] Manuele Rusci, Alessandro Capotondi, and Luca Benini. 2020. Memory-Driven Mixed Low Precision Quantization for Enabling Deep Network Inference on Microcontrollers. In *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze (Eds.), Vol. 2. 326–335.

[82] M. Ouzzani S. Thirumuruganathan N. Tang and A. Doan. 2020. Data Curation with Deep Learning. In *EDBT*.

[83] Emad W Saad and Donald C Wunsch II. 2007. Neural network explanation using inversion. *Neural networks* 20, 1 (2007), 78–93.

[84] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. 2020. Toward interpretable machine learning: Transparent deep neural networks and beyond. *arXiv preprint arXiv:2003.07631* (2020).

[85] Sebastian Schelter, Yuxuan He, Jatin Khilnani, and Julia Stoyanovich. 2019. FairPrep: Promoting Data to a First-Class Citizen in Studies on Fairness-Enhancing Interventions. *arXiv preprint arXiv:1911.12587* (2019).

[86] Thibault Sellam, Kevin Lin, Ian Huang, Michelle Yang, Carl Vondrick, and Eugene Wu. 2019. DeepBase: Deep Inspection of Neural Networks. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 1117–1134. https://doi.org/10.1145/3299869.3300073

[87] Jaydeep Sen, Fatma Ozcan, Abdul Quamar, Greg Stager, Ashish Mittal, Manasa Jammi, Chuan Lei, Diptikalyan Saha, and Karthik Sankaranarayanan. 2019. Natural Language Querying of Complex Business Intelligence Queries. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 1997–2000. https://doi.org/10.1145/3299869.3320248

[88] Dinggang Shen, Guorong Wu, and Heung-Il Suk. 2017. Deep Searning in Medical Image Analysis. *Annual review of biomedical engineering* 19 (2017).

[89] Michael Skirpan and Micha Gorelick. 2017. The Authority of" Fair" in Machine Learning. *arXiv preprint arXiv:1706.09976* (2017).

[90] Brad Smith. 2020. Microsoft will be carbon negative by 2030. (2020). https://blogs.microsoft.com/blog/2020/01/16/microsoft-will-be-carbon-negative-by-2030/

[91] Sebastian U. Stich. 2019. Local SGD Converges Fast and Communicates Little. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1g2JnRcFX

[92] Julia Stoyanovich, Serge Abiteboul, and Gerome Miklau. 2016. Data, Responsibly: Fairness, Neutrality and Transparency in Data Analysis. International Conference on Extending Database Technology.

[93] Julia Stoyanovich and Bill Howe. 2019. Nutritional Labels for Data and Models. *Data Engineering* (2019), 13.

[94] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. 2019. Mnasnet: Platform-aware neural architecture

search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2820–2828.

[95] P. Teich. 2018. Tearing Apart Google's TPU 3.0 AI Coprocessor. (2018). https://www.nextplatform.com/2018/05/10/tearing-apart-googles-tpu-3-0-ai-coprocessor/

[96] Joshua B Tenenbaum, Vin De Silva, and John C Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *science* 290, 5500 (2000), 2319–2323.

[97] Erico Tjoa and Cuntai Guan. 2020. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[98] R. Toews. 2020. Deep Learning's Carbon Emissions Problem. (2020). https://www.forbes.com/sites/robtoews/2020/06/17/deep-learnings-climate-change-problem/?sh=56e24b256b43

[99] Luan Tran, Min Y. Mun, Matthew Lim, Jonah Yamato, Nathan Huh, and Cyrus Shahabi. 2020. DeepTRANS: A Deep Learning System for Public Bus Travel Time Estimation Using Traffic Forecasting. *Proc. VLDB Endow.* 13, 12 (Aug. 2020), 2957–2960. https://doi.org/10.14778/3415478.3415518

[100] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[101] Manasi Vartak, Joana M F. da Trindade, Samuel Madden, and Matei Zaharia. 2018. Mistique: A system to store and query model intermediates for model diagnosis. In *Proceedings of the 2018 International Conference on Management of Data*. 1285–1300.

[102] Tin Vu. 2019. Deep Query Optimization. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 1856–1858. https://doi.org/10.1145/3299869.3300104

[103] Robert Walton. 2020. Google machine learning shifts data center operations to maximize efficiency, renewables use. (2020). https://www.utilitydive.com/news/google-machine-learning-shifts-data-center-operations-to-maximize-efficienc/577074/

[104] Zeke Wang, Kaan Kara, Hantian Zhang, Gustavo Alonso, Onur Mutlu, and Ce Zhang. 2019. Accelerating Generalized Linear Models with MLWeaving: A One-Size-Fits-All System for Any-Precision Learning. *Proc. VLDB Endow.* 12, 7 (March 2019), 807–821. https://doi.org/10.14778/3317315.3317322

[105] Abdul Wasay, Manos Athanassoulis, and Stratos Idreos. 2015. Queriosity: Automated Data Exploration. In *Proceedings of the IEEE International Congress on Big Data*. 716–719. https://doi.org/10.1109/BigDataCongress.2015.116

[106] Abdul Wasay, Brian Hentschel, Yuze Liao, Sanyuan Chen, and Stratos Idreos. 2020. MotherNets: Rapid Deep Ensemble Learning. In *Proceedings of the 3rd MLSys Conference (MLSys)*.

[107] Abdul Wasay and Stratos Idreos. 2021. More or Less: When and How to Build Convolutional Neural Network Ensembles. In *International Conference on Learning Representations*.

[108] Abdul Wasay, Xinding Wei, Niv Dayan, and Stratos Idreos. 2017. Data Canopy: Accelerating Exploratory Statistical Analysis. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 557–572. http://doi.acm.org/10.1145/3035918.3064051

[109] Wired. 2019. The Apple Card Didn't 'See' Gender—and That's the Problem. (2019). https://www.wired.com/story/the-apple-card-didnt-see-genderand-thats-the-problem/

[110] Shuang Wu, Guoqi Li, Feng Chen, and Luping Shi. 2018. Training and Inference with Integers in Deep Neural Networks. In *International Conference on Learning Representations*.

[111] Hongxia Yang. 2019. AliGraph: A Comprehensive Graph Neural Network Platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 3165–3166. https://doi.org/10.1145/3292500.3340404

[112] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).

[113] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. *Proceedings of the British Machine Vision Conference* (2016).

[114] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.

[115] Ji Zhang, Yu Liu, Ke Zhou, Guoliang Li, Zhili Xiao, Bin Cheng, Jiashu Xing, Yangtao Wang, Tianheng Cheng, Li Liu, Minwei Ran, and Zekang Li. 2019. An End-to-End Automatic Cloud Database Tuning System Using Deep Reinforcement Learning. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 415–432. https://doi.org/10.1145/3299869.3300085

[116] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep Mutual Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[117] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4320–4328.