Data Collection and Quality Challenges for Deep Learning

Steven Euijong Whang[*]
KAIST
swhang@kaist.ac.kr

Jae-Gil Lee
KAIST
jaegil@kaist.ac.kr

ABSTRACT

Software 2.0 refers to the fundamental shift in software engineering where using machine learning becomes the new norm in software with the availability of big data and computing infrastructure. As a result, many software engineering practices need to be rethought from scratch where data becomes a first-class citizen, on par with code. It is well known that 80–90% of the time for machine learning development is spent on data preparation. Also, even the best machine learning algorithms cannot perform well without good data or at least handling biased and dirty data during model training. In this tutorial, we focus on data collection and quality challenges that frequently occur in deep learning applications. Compared to traditional machine learning, there is less need for feature engineering, but more need for significant amounts of data. We thus go through state-of-the-art data collection techniques for machine learning. Then, we cover data validation and cleaning techniques for improving data quality. Even if the data is still problematic, hope is not lost, and we cover fair and robust training techniques for handling data bias and errors. We believe that the data management community is well poised to lead the research in these directions. The presenters have extensive experience in developing machine learning platforms and publishing papers in top-tier database, data mining, and machine learning venues.

PVLDB Reference Format:
Steven Euijong Whang, Jae-Gil Lee. Data Collection and Quality Challenges for Deep Learning. *PVLDB*, 13(12): 3429-3432, 2020.
DOI: https://doi.org/10.14778/3415478.3415562

1. OVERVIEW

Software 2.0 is a fundamental paradigm shift in software engineering. Recently, machine learning is becoming the mainstream due to the availability of big data and powerful computing infrastructure. The key characteristics are 1) data becomes a first class citizen, on par with code and 2) machine learning models become the new software. As a result, we need to rethink how to develop software.

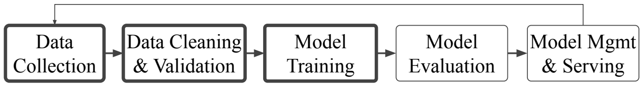

Figure 1: End-to-end Deep Learning. We will cover data collection and quality challenges in the first three steps from data collection to model training.

It is well known that 80–90% of the time spent on machine learning development is data preparation. Unfortunately, the opposite effort is spent on machine learning algorithms instead of the actual bottleneck [16]. Also, data quality has a profound impact on model accuracy where even the best machine learning algorithms cannot perform well without good data or at least handling dirty data during training.

In this tutorial, we investigate data quality challenges that occur in deep learning. Figure 1 shows the end-to-end process starting from data collection to model serving. In comparison to traditional machine learning, feature engineering is less of a concern, but there is instead a need for large amounts of training data. Unfortunately, the lack of data is one of the main reasons many industries are reluctant to adopt deep learning [16] (the other reason being lack of explainability). We thus focus on how to collect sufficient amounts of data. In addition, we need to validate and clean the data. While there is a vast literature on data cleaning, not all of the techniques are beneficial to machine learning [8]. In addition, recent machine learning issues including data poisoning need to be addressed as well. Even after carefully preparing the data, the data quality may still be problematic, and we need to cope with biased, dirty, or missing data using fair and robust model training [14, 15].

As the role of the data management community for deep learning grows rapidly, we believe this tutorial is timely and will discuss the opportunities to which we can contribute.

Scope and Structure. We will cover the first three steps in end-to-end deep learning (Figure 1). Section 2 provides a comprehensive view of data collection for machine learning. Section 3 presents data validation and cleaning techniques for machine learning. Section 4 presents model training techniques for coping with biased and dirty data.

Target Audience and Background. We target deep learning users that need an overview of how to collect data, en-

[*]Supported by a Google AI Focused Research Award.

Proceedings of the VLDB Endowment, Vol. 13, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3415478.3415562

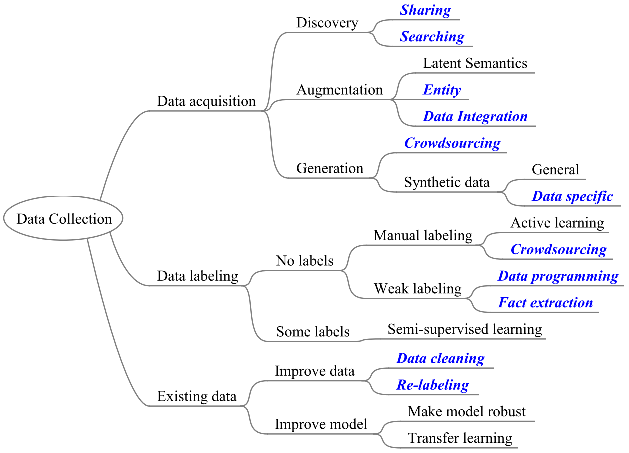

Figure 2: Data collection for machine learning [13]. The techniques in the leaf nodes that are at least partially proposed by the data management community are highlighted in italic blue font. A key observation is that there is an convergence of techniques between the data management and machine learning communities, so one needs to know both sides to understand the overall research landscape.

sure its quality, and cope with any data quality issues during model training. The audience needs to have a data management background plus working knowledge in deep learning.

Tutorial Length. This tutorial is intended for 3 hours. However, if the timeslot is not available, we can go for 1.5 hours; then, Sections 2 and 3 will be shortened while keeping the scope and structure.

2. DATA COLLECTION

Our coverage of data collection is based on a recent survey [13] by a presenter. According to Figure 2, there are largely three methods for data collection. First, data acquisition is the problem of finding the right datasets for training models. For example, as the number of datasets around us is increasing rapidly, searching for the right ones itself becomes a challenge. Second, data labeling is necessary in all supervised learning applications. Since manual labeling can be expensive, various scalable techniques have been proposed using semi-supervised learning, crowdsourcing, and weak supervision. Finally, one can also improve the quality of existing data or use transfer learning to re-use existing models instead of training from scratch.

2.1 Data Acquisition

Data acquisition is the process of finding datasets that are suitable for training machine learning models. In this tutorial we cover three approaches: data discovery, data augmentation, and data generation. Data discovery is the problem of indexing and searching datasets that may exist in corporate data lakes or the Web. Data augmentation is related, but focuses on complementing existing datasets by integrating them with external data. If there is not enough data around, the last resort is to take matters in one's hand and create datasets using crowdsourcing or synthetic data generation techniques.

2.2 Data Labeling

Once there are enough datasets, the next step is to label the examples. For example, in a smart factory application, one may want to label images of products on whether they are defective or not. The traditional approach for labeling is semi-supervised learning where the idea is to use existing labels to predict the other labels as much as possible. Other approaches include crowdsourcing where workers manually label examples or more advanced techniques like active learning where the manual labeling is done only for examples that are most likely to benefit the model's accuracy. Most recently, weak supervision is on the rise where the idea is to (semi-)automatically generate labels that are not perfect (therefore called "weak" labels), but at scale where the larger volume may compensate for the lower label quality. Weak supervision is useful in applications where there are few or no labels to start with.

2.3 Improving Existing Data and Models

In addition to searching and labeling datasets, one can also improve the quality of existing data and models. This approach is useful in two scenarios. If the application is novel or non-trivial where there are no relevant datasets outside, then the only choice is to make the best of what is already available. In other cases, collecting more data may no longer benefit the model's accuracy due to its low quality. Here the better options may be to clean the data or perform re-labeling to correct any mistakes. Yet another approach is to make the model training more robust to dirty data, a topic we cover in Section 4.2. In addition to improving data, one can also re-use existing models using transfer learning techniques.

3. DATA VALIDATION AND CLEANING

It is common for the training data to contain various errors. Machine learning platforms like TensorFlow Extended (TFX) [1] have data validation components to detect such data errors in advance using data visualization and schema generation techniques. Data cleaning can be used to actually fix the data, and there is a heavy literature [6] on various integrity constraints. However, recent studies [8] show that only focusing on improving the data does not necessarily benefit machine learning accuracy. In addition, in a machine learning point of view, we also need to address critical issues including data sanitization against poisoning and model fairness, which are actively studied in the security and machine learning communities, respectively. While the solutions for data sanitization are similar to those for data cleaning, improving model fairness is usually done during model training and thus covered in Section 4.

3.1 Data Validation

While there is a plethora of data visualization techniques, we focus on the ones that are most relevant to machine learning. Facets, a component of TFX, shows various statistics of datasets that are relevant for machine learning. More advanced tools include SeeDB [17], which can repeatedly generate possible visualizations that are of interest. This approach has the problem of false positives, so hypothesis testing started to be used in systems like CUDE [19] to guarantee the statistical significance of the findings.

Data validation focuses on finding problems in the data that affect the machine learning pipeline. TensorFlow Data

Validation [2] automatically generates database schemas from previous datasets and validates future datasets with the schema. There are largely three types of data errors that are validated. First, the data may be dirty; e.g., there may be duplicate country codes that are in upper and lower case letters. Second, the data may change as the data source itself evolves; e.g., the unit of a numeric feature may change from days to hours. Third, the data may simply be missing due to possible bugs in the data source; e.g., the title information of documents may be missing for a large portion of examples. For each schema violation, the user can either fix the data or the schema itself.

3.2 Data Cleaning

Data cleaning has a long history of removing various well-defined errors by satisfying integrity constraints including key constraints, domain constraints, referential integrity constraints, and functional dependencies. Unfortunately, only focusing on fixing the data does not necessarily guarantee the best model accuracy. We cover the recent CleanML [8] work, which systematically studies the impact of data cleaning on the accuracy of the model trained on that data. The conclusions are twofold: data cleaning does not necessarily improve the model accuracy, and performing model selection can at least reduce any negative effects where the data cleaning may harm model accuracy. Hence, we cover recent data cleaning techniques that are specifically geared towards improving model accuracy.

3.3 Data Sanitization

Data poisoning has recently become a serious issue because changing a fraction of the training data, which may come from an untrusted source, may alter the model's behavior. Compared to dirty data, there is a malicious intention of making the model fail. Early work focused on specific applications like spam detection and sensors. More recent studies are more general, but still tend to focus on specific models. It is unclear if there will be anything close to a unifying solution. The notion of data sanitization was introduced in 2008 [4] where attacks were assumed to occur in relatively confined time intervals, and the sanitization techniques used training metadata. More recently, adversarial machine learning, which attempts to fool models through malicious inputs (e.g., adversarial images), has become one of the most popular topics in machine learning.

4. MODEL TRAINING

Even after collecting the right data and cleaning it, data quality may still be an issue during model training. We present three directions of research. First, we cover fair model training techniques that can address data bias, which results in discriminative model behavior. Next, we cover robust model training techniques that cope with dirty data and still produce accurate results. Finally, we explore representation learning techniques for transforming the data in the best embedding format for model training.

4.1 Fair Training

Model fairness is becoming a critical issue where bias in the data may result in discriminatory behavior by the model. A famous example is the COMPAS tool by Northpointe, which predicts a defendant's risk of committing another crime. According to an analysis by ProPublica, black defendants are far more likely to be judged as a high risk compared to white defendants, which turns out to be inaccurate in practice. This investigation fueled the new research area of algorithmic fairness. Many definitions of fairness exist (see the survey [18]) and can be categorized as group or individual measures. Group fairness measures usually compare different sensitive groups (e.g., men versus women) and make sure their statistics are similar. For example, demographic parity is one of the popular measures that ensures the positive prediction rates of a model is similar among the sensitive groups. Government agencies use this fairness notion to ensure employers do not discriminate certain demographics when hiring. Individual fairness measures ensure that a prediction of an individual does not deviate too much from that of a similar person.

More recently, another line of research is to mitigate the unfairness, which can be done in largely three places: before model training on the data (pre-processing), during model training (in-processing), and after model training (post-processing). All of these approaches typically have a trade-off between accuracy and fairness, as the two objectives do not necessarily align. In this tutorial, we focus on pre-processing and in-processing approaches where examples are re-weighted to improve model fairness. For example, demographic parity can be improved by increasing the weights of positive examples of a sensitive group that has a lower positive prediction rate than other groups.

4.2 Robust Training

It is widely agreed that real-world datasets are dirty and erroneous despite the data cleaning process. For example, because data labeling is done manually in many cases, incorrect or missing labels are, in fact, very common; the proportion of incorrect labels is reported to be 8–38% in several real-world datasets [15]. Besides, especially in multivariate time-series data, missing values are unavoidable because of its high input rate and sensor malfunction. Thus, many deep learning techniques have been developed to consider the existence of data noises and errors, which are more critical in deep learning than in conventional machine learning as a deep neural network (DNN) completely memorizes such noises and errors because of its high expressive power.

Noisy or Missing Labels. Regarding noisy labels, recent techniques are mainly categorized into loss correction and sample selection. The former estimates the confidence of a label for each sample and adjusts the loss for the sample based on its label confidence during backward propagation. The latter also estimates the confidence of a label for each sample and includes the samples in training only if their label confidence is above some threshold. Recently, the sample selection approach becomes dominant, and a hybrid of the two approaches has been proposed [15]. Regarding missing labels, semi-supervised learning builds a model from a mixture of labeled and unlabeled data, by adopting unsupervised loss or collaborating with mix-up augmentation for unlabeled data. The representative techniques will be selectively covered in this tutorial.

Missing Data. Because missing data can reduce the statistical power and produce biased estimates, data imputation has been an active research topic in statistics and machine learning. In this tutorial, we focus on the deep learning techniques for time-series data. Because the recurrent neural network (RNN) is typically used for time-series data, the RNN family has been extended to receive the context about missing values. The most well-known technique is GRU-D [3] developed for medical data analysis, and many variations such as Temporal Belief Memory are available.

4.3 Representation Learning

It is common to transform or convert input data to improve its learning suitability, more precisely, to have the best representation of the data for learning a DNN model. This technique is collectively called *data emebedding*. For example, when a location is fed to a DNN model, a coordinate in a high-dimensional embedding space is used instead of the original latitude and longitude coordinate [11]. That is, this embedding achieves high enough dimensionality to have the descriptive capacity in the DNN model. The best embedding is usually obtained in an end-to-end training procedure. Recently, the embedding quality has improved further by considering the power-law characteristics of real-world datasets [9]. In this tutorial, we will focus on the embedding of spatial data because that of text data has been covered in natural language processing conferences.

5. BIOGRAPHIES OF PRESENTERS

Steven Euijong Whang is an assistant professor at the School of Electrical Engineering and Graduate School of AI, KAIST. His research interests are big data - AI Integration, big data analytics, and big data systems. Previously he was a Research Scientist at Google Research and co-developed the data infrastructure of the TensorFlow Extended (TFX) end-to-end machine learning platform. He received his Ph.D. in computer science in 2012 from Stanford University where his thesis topic was on data quality (entity resolution). He is a recipient of the Google AI Focused Research Award in 2018, the first in Asia.

Jae-Gil Lee is an associate professor at the Graduate School of Knowledge Service Engineering, KAIST. Before joining KAIST in 2010, he worked at the IBM Almaden Research Center and the University of Illinois Urbana-Champaign. He earned his Ph.D. in computer science in 2005 from KAIST. His research interests encompass spatio-temporal data mining and scalable machine learning, and he is recently working on the data quality issues for deep learning. He received the Best Paper Award at AAAI ICWSM 2013. He is serving as an associate editor of IEEE TKDE and a steering committee member of PAKDD since 2019.

6. OTHER VENUES AND TUTORIALS

Steven was a co-presenter of the ACM SIGMOD 2017 tutorial "Data Management Challenges for Production Machine Learning" [12]. Another tutorial on the same topic, but with a different perspective, was presented concurrently [7]. In comparison, at least two thirds of this tutorial is completely new. In particular, Sections 2 and 4 are new, and Section 3 covers recent issues in data cleaning for machine learning. Other related tutorials include data cleaning [5] and data lake management [10], presented in VLDB 2018 and 2019, respectively. Our tutorial is a natural follow-up that covers the entire spectrum of data collection and addresses data quality issues in deep learning.

7. REFERENCES

[1] D. Baylor, E. Breck, H. Cheng, et al. TFX: A tensorflow-based production-scale machine learning platform. In *KDD*, pages 1387–1395, 2017.

[2] E. Breck, M. Zinkevich, N. Polyzotis, S. Whang, and S. Roy. Data validation for machine learning. In *MLSys*, 2019.

[3] Z. Che, S. Purushotham, K. Cho, D. A. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 2018.

[4] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis. Casting out demons: Sanitizing training data for anomaly sensors. In *IEEE S&P*, pages 81–95, 2008.

[5] L. Dong and T. Rekatsinas. Data integration and machine learning: A natural synergy. *PVLDB*, 11(12):2094–2097, 2018.

[6] I. F. Ilyas and X. Chu. *Data Cleaning*. ACM, 2019.

[7] A. Kumar, M. Boehm, and J. Yang. Data management in machine learning: Challenges, techniques, and systems. In *SIGMOD*, pages 1717–1722, 2017.

[8] P. Li, X. Rao, J. Blase, Y. Zhang, X. Chu, and C. Zhang. CleanML: A benchmark for joint data cleaning and machine learning. *CoRR*, abs/1904.09483, 2019.

[9] X. Li, K. Zhao, G. Cong, C. S. Jensen, and W. Wei. Deep representation learning for trajectory similarity computation. In *ICDE*, pages 617–628, 2018.

[10] F. Nargesian, E. Zhu, R. J. Miller, K. Q. Pu, and P. C. Arocena. Data lake management: Challenges and opportunities. *PVLDB*, 12(12):1986–1989, 2019.

[11] D. Park, H. Song, M. Kim, and J. Lee. TRAP: Two-level regularized autoencoder-based embedding for power-law distributed data. In *TheWebConf*, 2020.

[12] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich. Data management challenges in production machine learning. In *SIGMOD*, pages 1723–1726, 2017.

[13] Y. Roh, G. Heo, and S. E. Whang. A survey on data collection for machine learning: a big data - AI integration perspective. *IEEE TKDE*, 2019.

[14] Y. Roh, K. Lee, S. E. Whang, and C. Suh. FR-Train: A mutual information-based approach to fair and robust training. In *ICML*, 2020.

[15] H. Song, M. Kim, and J. Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In *ICML*, pages 5907–5915, 2019.

[16] M. Stonebraker and E. K. Rezig. Machine learning and big data: What is important? *IEEE Data Eng. Bull.*, 2019.

[17] M. Vartak, S. Rahman, S. Madden, A. G. Parameswaran, and N. Polyzotis. SEEDB: Efficient data-driven visualization recommendations to support visual analytics. *PVLDB*, 8(13):2182–2193, 2015.

[18] S. Venkatasubramanian. Algorithmic fairness: Measures, methods and representations. In *PODS*, page 481, 2019.

[19] Z. Zhao, L. D. Stefani, E. Zgraggen, C. Binnig, E. Upfal, and T. Kraska. Controlling false discoveries during interactive data exploration. In *SIGMOD*, pages 527–540, 2017.