Towards a Generic Deep Learning Pipeline for Traffic Measurements

Raphaël Azorin
Huawei Technologies, France

Massimo Gallo
Huawei Technologies, France

Alessandro Finamore
Huawei Technologies, France

Maurizio Filippone
Eurecom, France

Pietro Michiardi
Eurecom, France

Dario Rossi
Huawei Technologies, France

CCS CONCEPTS

• **Networks** → **Network measurement**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

network measurements, deep learning, representation learning

ACM Reference Format:
Raphaël Azorin, Massimo Gallo, Alessandro Finamore, Maurizio Filippone, Pietro Michiardi, and Dario Rossi. 2021. Towards a Generic Deep Learning Pipeline for Traffic Measurements. In *CoNEXT Student Workshop 2021 (CoNEXT-SW '21), December 7, 2021, Virtual Event, Germany.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3488658.3493785

1 INTRODUCTION

Traffic measurements are key for network management as testified by the rich literature from both academia and industry. At their foundation, measurements rely on transformation functions $f(x) = y$, mapping input traffic data x to an output performance metric y. Yet, common practices adopt a *bottom-up* design (i.e., metric-based) which leads to (*i*) invest a lot of efforts into (re)discovering how to perform such mapping and (*ii*) create specialized solutions. For instance, sketches are a compact way to extract traffic properties (heavy-hitters, super-spreaders, etc.) but require analytical modeling to offer correctness guarantees and careful engineering to enable in-device deployment and network-wide measurements.

Rather than relying on network experts domain knowledge, Machine Learning offers algorithms to learn $f(x) = y$ mappings. In particular, Deep Neural Networks (DNNs) act as universal functions approximators as they can learn complex input-output data relationships with automatic feature extraction. Thus, contextualized to traffic measurements, Deep Learning (DL) could empower a *top-down* design (i.e., model-based), potentially fostering automation and generalization. Yet, we claim that its application to traffic measurements is still in its infancy and we identified two research questions that require more work in the community.

First, considering the nature of network traffic, Q_1: *What is the appropriate representation of network traffic to facilitate knowledge extraction with DL?* Despite being automatic, the feature extraction operated by DL algorithms highly depends on the input traffic representation. Thus, we need to adapt the DL workflow to the specific characteristics of traffic data [4]. For example, images used in Computer Vision (CV) are represented as static grids of fixed size. Such encoding is not transferable to networking which deals with sequential data of arbitrary size.

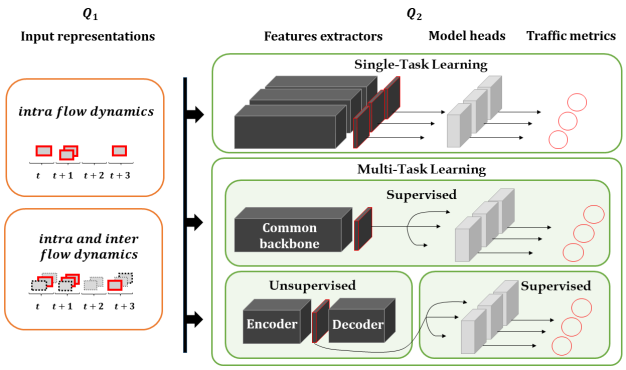

Figure 1: Schematic pipelines to compare some Deep Learning approaches for N traffic measurements.

Second, considering generalization, Q_2: *Can we design a unified DL pipeline that serves several traffic measurements?* CV and Natural Language Processing demonstrated that it is possible to reuse DNNs knowledge across related tasks (e.g., Transfer Learning). Alternatively, the feature extraction process can be designed to learn several mappings at the same time (e.g., Multi-Task Learning). In the context of networking, this would allow operators to derive multiple measurements from a single common representation of the input traffic, thus reducing the burdens of managing several individual models. However, often this leads to trade off performance for generalization.

To answer these two questions, we propose an empirical campaign to study a variety of modeling approaches for traffic metrics prediction.

2 BACKGROUND

Traffic representation for Deep Learning. A DL model approximates a function mapping an input x to an output y. Under the hood, this mapping is operated via a double transformation. First, the raw data x is projected into a latent space by the model *backbone*, which does most of the knowledge extraction. Second, points in the latent space are transformed into the final output y thanks to the model *head*, which is typically a simple linear layer. These two transformations compose the $f(x) = y$ mapping. Conceptually, the latent representation is expected to capture latent properties in the input data. The representation effectiveness depends not only on the DNN architecture but also on the input data fed during

CoNEXT-SW '21, December 7, 2021, Virtual Event, Germany

training. Specifically, the input should be presented in a way that stresses the salient characteristics deemed helpful for the mapping. For example, images in CV are represented by a grid of pixels, allowing Convolutional Neural Networks to extract patterns at different scales in the picture (low-level, mid-level and high-level features by exploiting spatial locality). Similarly, in networking, traffic patterns may emerge when capturing properties across several packets. Yet, common traffic representations focus on encoding packet-level information (e.g., one-hot encoding of packet headers features [2]) or flow-level information (e.g., time series of packet inter-arrival time or size [5]). We argue that those representations fail to capture both intra-flow and inter-flow relationships: analyzing a packet in isolation is likely insufficient to understand a flow; likewise, flow dynamics are influenced by other flows. Thus, Q_1 calls for a deeper investigation of the design space.

Deep Learning pipelines. Q_2 suggests that parts of the learning process might be shared across different measurements. At a high level, we identify three "pipeline" designs (cf. Figure 1 - right). The baseline approach consists in a single pair of backbone and head. Training is guided by a loss minimization objective between a single target measurement and a predicted one. To this extent, this kind of supervised DNNs act as feature extractors for *Single-Task Learning (STL)*. In order to amortize training and maintenance costs, operators may be interested in mutualizing the learning process across several related measurement tasks. To accommodate this need, we identify two possible pipeline designs for such *Multi-Task Learning (MTL)*. Both produce a single representation common to several tasks, but they differ in the way this representation is learned: in a supervised or an unsupervised fashion.

In a supervised MTL approach, the training process considers several model heads plugged on the same model backbone. Each head takes care of a specific target mapping and the latent space construction is guided by concurrent supervised learning objectives. The intuition behind this approach is that one task may benefit from the knowledge learned during the training of other related tasks. However, this is not a guarantee and this knowledge transfer can even be detrimental to some tasks [1].

The unsupervised MTL approach requires a two-step process. First, the learning objective is focused on parametrizing an encoder and a decoder that optimize input reconstruction, e.g., (Variational) Auto-Encoders. Once the encoder/decoder pair has been trained, the decoder is replaced by several model heads that will map the same encoded representation to several metrics. This second step of the pipeline training is thus conducted in a supervised setting and the encoder may or may not be adjusted (fine-tuning) for the metrics to model.

3 METHODOLOGY

Our goal is to evaluate how specific approaches compare to more generic ones by studying the trade-off between efficiency and generality. To do so, we aim to assess various strategies in terms of accuracy and cost for a set of traffic measurements. As a starting point, we consider the three approaches detailed in Section 2, with two variations for input traffic representation. We focus our study on three example flow-related measurements: two related regression tasks (flow completion time and average round trip time) and one classification task (traffic classification).

Input pre-processing. Input features related to the target measurements (e.g., packets inter-arrival time, headers values) need to be presented in a way that preserves flow inter- and intra-relationships. As sketched in Figure 1 - left, in order to capture a flow's internal dynamics, its packets are batched in consecutive time windows. A second option adds packets from concurrent flows to the previously defined batches, thus also enabling the capture of inter-flow dynamics. Results comparison between these two options will help answer Q_1.

DNNs architectures. Out of the three approaches we consider (cf. Figure 1 - right), the STL pipeline is composed of three distinct supervised DNNs, one per metric. The fully supervised MTL pipeline consists of a single backbone and three heads, one per metric. In this case, all the model's components are trained together at once with a common objective expressed as a weighted linear combination of the distinct losses [3]. For the encoder/decoder-based MTL pipeline, training is first performed without supervision. In a second step, the latent representation in the bottleneck is forwarded to three models heads which can be trained separately at a later time, or concurrently with the encoder. Comparison of these approaches in terms of accuracy, training cost and inference cost will help answer Q_2.

4 FUTURE EVALUATION

To evaluate and compare the proposed approaches, we consider traffic traces (PCAP) extracted from multiple vantage points in the network for a given time window. We plan to first use simulated data (e.g., generated with NS3) to test our method on a simple network and then adapt it to real-world traces. To assess the final performance of each approach, we propose to relate the accuracy of their measurement to their task cost. Accuracy can be measured with standard metrics depending of the task at hand: e.g., Mean Squared Error for the regression tasks and F1-score for the classification task. As for the task cost, it can be interpreted in many ways: total number of model parameters, number of training epochs [1], total training time, or even inference time. These metrics measure different cost aspects that need to be weighted in accordance with the network operator's priorities. Eventually, as stressed in [1], all tasks may not learn at the same rate, which is why each competitor may require a different training budget.

REFERENCES

[1] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
[2] Jordan Holland, Paul Schmitt, Nick Feamster, and Prateek Mittal. 2020. nprint: A standard data representation for network traffic analysis. *arXiv preprint arXiv:2008.02695* (2020).
[3] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. *CoRR* abs/1810.04650 (2018). arXiv:1810.04650
[4] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang. 2018. Machine Learning for Networking: Workflow, Advances and Opportunities. *IEEE Network* 32, 2 (2018), 92–99. https://doi.org/10.1109/MNET.2017.1700200
[5] Kun Yang, Samory Kpotufe, and Nick Feamster. 2020. A Comparative Study of Network Traffic Representations for Novelty Detection. *arXiv preprint arXiv:2006.16993* (2020).