Neural Network Meets DCN: Traffic-driven Topology Adaptation with Deep Learning

Mowei Wang, Yong Cui
Tsinghua University

Shihan Xiao
Huawei Technologies

Xin Wang
Stony Brook University

Dan Yang
Beijing University of Posts and Telecommunications

Kai Chen
Hong Kong University of Science and Technology

Jun Zhu
Tsinghua University

ABSTRACT

The emerging optical/wireless topology reconfiguration technologies have shown great potential in improving the performance of data center networks. However, it also poses a big challenge on how to find the best topology configurations to support the dynamic traffic demands. In this work, we present *xWeaver*, a traffic-driven deep learning solution to infer the high-performance network topology online. xWeaver supports a powerful network model that enables the topology optimization over different performance metrics and network architectures. With the design of properly-structured neural networks, it can automatically derive the critical traffic patterns from data traces and learn the underlying mapping between the traffic patterns and topology configurations specific to the target data center. After offline training, xWeaver generates the optimized (or near-optimal) topology configuration online, and can also smoothly update its model parameters for new traffic patterns. The experiment results show the significant performance gain of xWeaver in supporting smaller flow completion time.

CCS CONCEPTS

• **Networks** → **Topology analysis and generation**; **Data center networks**; • **Computing methodologies** → *Machine learning*;

KEYWORDS

Data Center Networks; Topology Adaptation; Deep Learning

ACM Reference Format:

Mowei Wang, Yong Cui, Shihan Xiao, Xin Wang, Dan Yang, Kai Chen, and Jun Zhu. 2018. Neural Network Meets DCN: Traffic-driven Topology Adaptation with Deep Learning. In *SIGMETRICS '18 Abstracts: ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems Abstracts, June 18–22, 2018, Irvine, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3219617.3219656

1 INTRODUCTION

Data center network (DCN) is the key infrastructure of cloud computing. With the fast growth of cloud services, the traffic in today's DCNs shows high spatial-temporal dynamics. To address this issue at low cost, recent work proposes to construct *topology-reconfigurable* DCNs by introducing the new network components such as optical circuit switches (OCS) or wireless radios into the DCNs [2, 3, 5, 6]. Rather than relying on heavy network overprovision, the agile optical/wireless links can be quickly switched to construct a proper runtime topology to meet the current traffic demands.

The key challenge in supporting the topology-reconfigurable architectures is how to obtain the optimal (or near-optimal) topology configuration for the given traffic demands. Focusing on the *local* link configuration for an OCS switch based on its port demands [5], previous work generally ignores the interactions between OCS switches and the wired network topology. Recent studies [1, 7] show that it is beneficial to adapt the link configuration of an OCS switch along with a single electrical switch. Inspired by these observations, the goal of this work is to construct the best *global* topology to meet the traffic demands in a practical DCN.

A straightforward method is to model the global interactions between traffic and topology with respect to the optimization objective. However, this is non-trivial as the transmission performance of a topology is affected by many practical system factors, such as the specific routing protocols and congestion control strategies. The modeling becomes more difficult if using the higher-layer application performance as the optimization objective (e.g., the Hadoop job completion time). As shown in the previous work, even if we only consider the simplistic case with one OCS switch and one electrical switch, it often requires solving an integer linear programming (ILP) problem that is unscalable [6] due to the discrete property of topology configurations. The above modeling challenges drive existing work to resort to heuristic solutions that are simple, fast but potentially far away from optimal [3, 5, 6].

To address the above challenges, we present *xWeaver*, a traffic-driven deep learning system for the topology configuration in DCNs. The motivation is that the deep neural network can build up a comprehensive interaction model between traffic and topology automatically with little human efforts. State-of-the-art deep learning technologies, e.g., the convolutional neural network (CNN), are known to be good at learning complex models in image processing. Recently, they have made impressive performance breakthroughs in many fields. It is also a new trend to apply machine learning techniques on solving the networking problems [8].

Different from the conventional ILP-based modeling, in our system, the neural network does not need to keep solving a complex model online. Instead, in the offline phase, xWeaver uses a specialized neural network to learn and store the critical features of the

SIGMETRICS '18 Abstracts, June 18–22, 2018, Irvine, CA, USA

ACM ISBN 978-1-4503-5846-0/18/06.
https://doi.org/10.1145/3219617.3219656

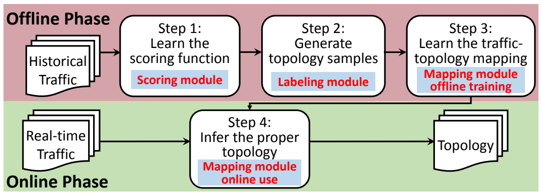

Figure 1: xWeaver framework overview

optimal (or near-optimal) solutions. The parameters of the neural network can be trained offline from the history traces, which are rich and easily available in today's DCNs. The parameters can also be updated with new data available. After the offline training, the topology inference through neural network is generally fast, especially supported by the speed-up using the advanced hardwares. Thus it can be easily applied for online topology configuration.

In xWeaver, we design two properly-structured neural networks to address the above questions (§2). Our simulations show that xWeaver can support higher flow performance than conventional solutions (§3). For the design details, extensive experiment results, and discussions, please refer to the full version of this paper [9].

2 XWEAVER FRAMEWORK

The overview of our xWeaver system is shown in Fig. 1. Generally, the mapping module of the xWeaver system provides the key function to infer the best topology from traffic demands, while the scoring module and labeling module are designed to support the efficient training of the mapping module. In the following, we will present the details of these basic modules respectively.

2.1 Scoring module

This module is used to provide a fast performance evaluation (denoted as scoring function) for any given pair of traffic demand matrix and topology (i.e., the *traffic-topology pair*). The module input is a traffic-topology pair and the output is a real-valued performance *score*. In xWeaver, we allow the network operators to define the score as any performance metric that they aim to optimize, e.g., the flow completion time or the upper-layer application metrics.

A straightforward method to implement the scoring function is to use a conventional network simulator (e.g., the ns-2) to mimic the real system settings. However, existing network simulators are relatively slow as they require simulating the detailed network protocols and packet-level transmission events, which is not feasible for our deep learning solution. The successful training of a neural network may requires more than 10000 score evaluations even with a small-scale DCN, which will result in a sample generation time as long as one month (minutes/sample). Hence in practice, we turn to find an approximate scoring function running much faster (milliseconds/sample) but within a tolerable accuracy loss.

The above issue motivates us to design a specialized neural network (SCNN) that has low complexity and high training efficiency. The key insight is that a pattern that emerges in the input traffic will hardly emerge in the input topology. Hence as Fig. 2 shows, we propose to use two separate multi-layer *convolutional* neural networks as the basic components to perform the feature extraction on traffic demand and topology configuration independently. This separate pre-process contributes to a high learning efficiency because each input only focuses on its own feature extraction. Moreover, the separate structure will generate a neural network with

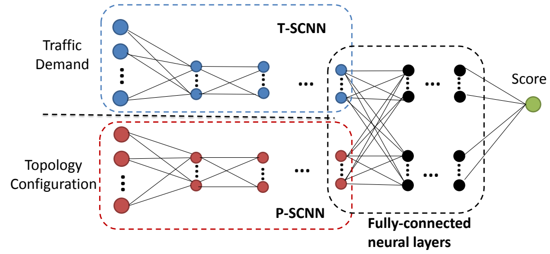

Figure 2: SCNN structure for feature extraction and scoring

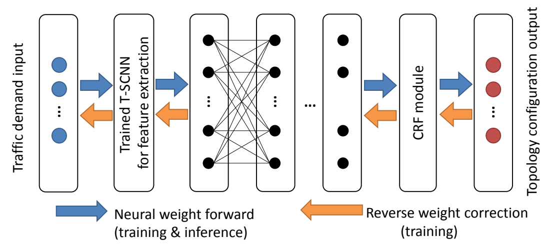

Figure 3: Structure design of the FPNN neural network

significantly smaller number of parameters for efficient training. Then at the end of the SCNN, we combine their derived feature results and map them to the final score with a small number of fully-connected neural layers. After the training, SCNN can be several orders of magnitude faster than directly running the network simulator. Our experiments also demonstrate that it can achieve a higher scoring accuracy and a faster convergence speed than the conventional fully-connected NN [9].

2.2 Labeling module

This module is designed to label historical traffic traces with corresponding topologies that have high performance scores. It provides the traffic-topology samples for later model training. Existing traffic traces generally do not have the information on the network topologies. The labeling is made possible and automatic with the facilitation of the *scoring module*. Specifically, the label for each traffic demand matrix can be obtained by searching the potential topology space with the *scoring module*.

We design a heuristic search algorithm to generate the high-score topology samples with a controllable time overhead. Let $N_\delta(p)$ denote the set of neighbor topologies of a given topology p, where each topology $\widehat{p} \in N_\delta(p)$ has at most δ different edges from the topology p. Here we also refer the parameter δ as the *search depth*. Consider a traffic trace $\{f_t\}$ collected from the target data center. For each traffic sample f_t at time t, we want to generate the topology sample p_t that satisfies: $p_t = \arg\max_{p \in N_\delta(p_{t-1})} Score(f_t, p)$. By limiting the search depth, we are able to search for a local-optimal solution within a reasonable period of time. However, with extensive simulations, the above direct search does not generate the satisfactory topology configurations. This is due to the fact that a lot of different neighbouring topologies are sharing similar scores. Thus it is inefficient to stop the search when finding only one local-optimal solution. In xWeaver, we use both the beam search and the random start schemes to help jump out of the local-optimal points. Thus it keeps exploring a better local-optimal solution until a pre-defined maximum number of iterations are reached.

2.3 Mapping module

This module is the core of xWeaver to learn the high-dimensional global mapping between traffic and topology. It consists of two processes: a) building a specialized neural network which maps a traffic demand matrix to a topology configuration, denoted as FPNN shown in Fig. 3; b) training the neural network parameters from the data samples provided by the *labeling module*. Then the trained neural network can be used to perform efficient topology inference online. We also provide a design option to enhance the output of FPNN by combining a flexible probability graph model CRF (Conditional Random Field). It can help explicitly embed prior human knowledge about the target network architecture (see [9]).

In FPNN, rather than using the raw traffic input to construct the mapping, we propose to exploit the critical features in the traffic input for dimensionality reduction. Specifically, we carefully select a part of the well-trained SCNN (denoted as the T-SCNN in Fig. 2) to extract the critical features from the input traffic. Then we copy both the structure and trained parameters of T-SCNN to construct the input of the neural layers of FPNN. The T-SCNN has been well trained in the scoring module and thus does not need to re-train from the scratch. After the feature extraction, we use a small number of fully-connected neural layers in FPNN to map the traffic features to the output topology configuration.

2.4 Workflow

By combining the two modules, the historical traffic traces can be *automatically labeled* with the corresponding high-score topologies without any human efforts. This automatic labeling process allows for generating enough labeled data for high performance training. These modules work only in the offline phase, so it can tolerate a relatively-long time to generate enough traffic-topology samples. After offline training from these samples, the mapping module can easily make online topology decisions in real time.

3 SIMULATION

Our simulations are conducted through a customized flow-level simulator. Since it is infeasible to obtain the optimal solutions when the network scale becomes large, we first focus on a DCN topology of a 4-port fat-tree. For the reconfigurable part, we follow the same OCS-based architecture setting in [2] and use a single 16-port OCS switch to connect all the ToR switches. Our optimization objective (i.e. the *score*) is defined as minimizing the completion time of the demands (CTD) [6]. We use a random traffic pattern with 20000 demand matrices as the traffic input. The comprehensive performance evaluation for each xWeaver module can refer to [9].

In xWeaver, we implement the two neural networks (SCNN and FPNN) with the deep-learning library `caffe`. We compare the topology performance of our xWeaver solution with three other solutions: 1) the *Weight-matching* solution, which generates the topology configuration with the Edmonds algorithm [4] used in existing topology-adaptation research work [2, 5]; 2) the *Sample* solution, where the topology configurations are generated our labeling module; 3) the *Optimal* solution, which has the lowest CTD for all the demands implemented by brute-force search.

In Fig. 4, the CTD of four solutions are evaluated on the test set. The CTD results are all normalized by the *Optimal* solution. We can see that although the performance of *Sample* varies quite a lot for different demands, our *xWeaver* solution trained from

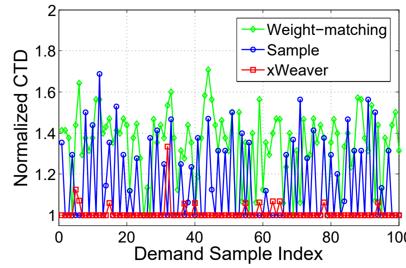

Figure 4: Performance comparison of different solutions.

Sample outperforms *Sample* in the CTD metric, and has its performance very close to that of the *Optimal* solution. *Sample* has a large fraction of high-quality topologies together with a small fraction of low-quality topologies. *xWeaver* shows the ability of learning the important demand patterns and key topology structures from the high-quality topology samples, and then using them to infer the high-quality topologies for new traffic demands. On the other hand, *Weight-matching* has the worst performance among all the solutions and experiences a large variation of topology qualities. Its completion time is about 50% higher than that of *xWeaver*.

4 CONCLUSION

In this paper, we present xWeaver, a traffic-driven deep learning solution that can enable the high-performance global topology configurations in DCNs with little human modeling efforts. Benefited from its automatic configuration features, xWeaver can support a variety of DCN architectures and the flexible optimization on self-defined objectives that the network operators prefer. Our experiments demonstrate that xWeaver achieves much shorter flow completion time than conventional solutions.

ACKNOWLEDGMENTS

This work is supported by the Science and Technology Project of State Grid Corporation of China (No.2017YFB1010002), the National 863 project (No. 2015AA015701) and the NSF grant (#CNS-1526843).

REFERENCES

[1] Shaileshh Bojja, Mohammad Alizadeh, and Pramod Viswanath. 2016. Costly Circuits, Submodular Schedules and Approximate Carathéodory Theorems. In *SIGMETRICS*.

[2] Kai Chen, Anubhav Singla, Ashutosh Singh, Kishore Ramachandran, Lei Xu, Yueping Zhang, Xitao Wen, and Yan Chen. 2014. OSA: an optical switching architecture for data center networks with unprecedented flexibility. In *NSDI*.

[3] Yong Cui, Shihan Xiao, Xin Wang, Zhenjie Yang, Chao Zhu, Xiangyang Li, Liu Yang, and Ning Ge. 2016. Diamond: nesting the data center network with wireless rings in 3D space. In *NSDI*.

[4] Jack Edmonds. 1965. Paths, trees, and flowers. *Canadian Journal of mathematics* 17, 3 (1965), 449–467.

[5] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2010. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *SIGCOMM*.

[6] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David" Wetherall. 2011. Augmenting data center networks with multi-gigabit wireless links. In *SIGCOMM*.

[7] He Liu, Matthew K Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M Voelker, David G Andersen, Michael Kaminsky, et al. 2015. Scheduling techniques for hybrid circuit/packet networks. In *CoNEXT*.

[8] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang. 2017. Machine Learning for Networking: Workflow, Advances and Opportunities. *IEEE Network* (2017).

[9] Mowei Wang, Yong Cui, Shihan Xiao, Xin Wang, Dan Yang, Kai Chen, and Jun Zhu. 2018. Neural Network Meets DCN: Traffic-driven Topology Adaptation with Deep Learning. *POMACS* (2018).