

Common Knowledge Based Transfer Learning for Traffic Classification

Yunming Xiao, Haifeng Sun, Zirui Zhuang, Jingyu Wang, Qi Qi

State Key Laboratory of Networking and Switching Technology

Beijing University of Posts and Telecommunications

Beijing, China

Email: {awdrtg, hfsun, zhuangzirui, wangjingyu, qiqi8266}@bupt.edu.cn

Abstract—Deep neural networks have been used for traffic classification, and promising results are obtained. However, most previous work confined to one specific classification task and restricted the classifiers potential performance and applications. As the traffic flows can be labeled from different perspectives, the performance of the classifier might be improved by exploring more meaningful latent features. For this purpose, we adopted a multi-output DNN model that simultaneously learns different traffic classification tasks. The common knowledge of traffic is exploited by the synergy among the tasks and boosts the individual performances of the tasks. Experiments show that this structure has the potential to meet new future demands and achieve the classification with advanced speeds and fair accuracies. Yet, due to the heavy training cost, the neural networks, though achieving good performance, are hard to implement in the real environment. We further show that few-shot learning could be a viable approach.

I. INTRODUCTION

With the promotion of hardware capacity and growing communication demands, the simple systems of the traditional internet network are gradually replaced with some new management systems, e.g. the SDN (software-defined network) [1]. A rising computation is implemented in the network. Hence, the network traffic classification has become one of the key points of the efficient network traffic management and QoS (Quality of Service) guarantees.

For the past several decades, many techniques were proposed to better predict the diverse requirements for bandwidth, latency, and quality of different flows. It is widely recognized that many properties of a flow can be well predicted with a small set of the flow features. Among the various approaches, some machine learning tools have shown promising results using the port number or some statistical features of the flows [2]. Recently, some studies have turned to deep learning techniques and have obtained good accuracies [3].

Our work was motivated by the fact that the demands will always be changing, and that most previous works have shown that some features can be used to discriminate among different tasks of flows. For instance, the features for classifying either elephant or mice flow and those for identifying the applications of flows are surprisingly similar [4]–[7]. This indicates that there might be some common knowledge of different traffic classification tasks and the mapping functions from the selected features to the different flow properties that we need to predict are closely related.

To explore the common knowledge of different flow classifications and come up with an adaptive solution, we compared many machine learning tools and selected the multi-output DNN structure, which is an adaptation of the simple DNN. Our adopted structure has two phases as follows: common knowledge is extracted at the first phase by the common layers, and then, the various flow classification tasks are concurrently trained and are computed with inter-independent branches of private layers. The common layers can effectively save calculation time and memory. Each of the classification tasks owns an extended and relatively simple branch of private layers for specialized tasks. Moreover, new related classification tasks can be solved by simply adding a new branch of private layers connected to the well-trained common layers.

The adopted structure might be a solution to the real-world implementation. Based on the common knowledge and the hierarchical structure, complex new tasks can be completed with only a small calculation cost of several neurons. Moreover, a new demand can be tested without influencing the established business and share the common knowledge extracted from a great amount of existing related data.

Our main contributions are summarized as follows:

- We adopted the multi-output DNN structure to extract the common knowledge of network traffic and a solution to the multi-task problems in traffic classification.
- We demonstrate that this structure can be capable of handling transfer learning for different related traffic identification tasks and thus meets our expectation of the potential to address new demands.
- Regarding the great cost of training neural networks and the difficulty to obtain specific datasets for some tasks, we further look into a possible real-world solution. We apply the proposed structure to few-shot learning scenarios. Experiments show notable performances on both accuracy and efficiency with our approach.

II. METHODOLOGY

A. Multi-Output DNN Approach

The mathematical intuition of the DNN is a highly nonlinear mapping function. Given a specific trace dataset and the flow properties $Y_1, Y_2, \dots, Y_n$ (e.g., flow rate), the corresponding mapping function $f_1, f_2, \dots, f_n$ may be similar in some

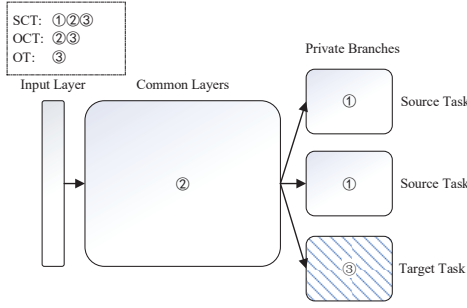


Fig. 1. The multi-output DNN structure and transfer training schemes. The SCT trains the source, common and target layers. The OCT trains only the common and target layers. The OT trains only the target layers.

degree with the same input X (i.e., flow features). With an allowable affordability deviation Δ_i , we have

$$f_i = f'_i \cdot f_c + \Delta_i. \quad (1)$$

f'_i refers to the mapping function of different branches of private layers and f_c denotes the general function of the common layers. Fig. 1 shows that a multi-output DNN contains common layers and several branches of private layers. All the common layers are fully connected, but there are no connections between neurons from different private branches.

Assuming that there are N total classification tasks, for each task (each branch of the neural network), we use a loss function $L_i[f_i(X, \theta), Y_i]$ to evaluate the performance of the corresponding fitting function where θ denotes the parameters in the neural networks. Hence, the optimization process is

$$\theta^* = \arg \min_{\theta=\{\theta_c, \theta_i\}} \sum_i^N L_i[(f'_i \cdot f_c)(X, \theta), Y_i]. \quad (2)$$

θ^* denotes the parameters of the final fitting function. θ_c and θ_i refer to the parameters in the common layers and i^{th} branch of the private layers respectively.

B. Transfer Learning

In some cases, a new label classification task is proposed, but it might be costly to train a new model. It may be practical to train the new classifier using the data from related tasks or using other tasks previously trained layers to speed up the training process. The new task is called the *target task* and the previously trained tasks are referred to the *source task*.

To achieve the transfer learning, the common layers should be trained, and the parameters in both the source and target tasks need to be optimized using backpropagation. We refer to this scheme to SCT (training source, common and target layers) in later sections, which is expressed as

$$\theta^* = \arg \min_{\theta=\{\theta_c, \theta_T\}} \sum_{j \in \{S, T\}} L_j[(f'_j \cdot f_c)(X, \theta), Y_j]. \quad (3)$$

One way to simplify the above equation is to ignore the calculation of previously trained tasks as follows:

$$\theta^* = \arg \min_{\theta=\{\theta_c, \theta_T\}} \sum_{j \in T} L_j[(f'_j \cdot f_c)(X, \theta), Y_j]. \quad (4)$$

TABLE I
STATISTICS OF THE DATASET

Trace	Start Time	Duration	Flow	App Num
ISPDLSL-I	10:54, Jan 6, 2009	1h 36min	23545	36
ISPDLSL-II	16:09, Jan 6, 2010	1h 21min	23892	50
Entry09	13:45, Aug 20, 2003	27min	66248	11
Entry10	14:55, Aug 20, 2003	27min	65036	11

Note that OCT (training only common and target layers) might negatively affect the source tasks.

Pursuing a more clear and simple function, the last method ignores the common layers.

$$\theta^* = \arg \min_{\theta_T} \sum_{j \in T} L_j\{f'_j[f_c(X, \theta_c), \theta_j], Y_j\} \quad (5)$$

As Eq. 5 illustrates, OT (training only target layers) concatenates the new task branch to the trained neural network but trains only the private layers of the new branch. In such an approach, the common layers will not be influenced. It meets the demand of testing a new task without influencing the established business. Fig. 1 illustrates a simplified version of the three training schemes.

C. Few-Shot Learning

The real-world traffic classification implementation of neural networks might be hard due to the heavy cost of training and it might be difficult to acquire large sets of training data. The few-shot learning can be a solution. In particular, a training set with ample data is adopted for the training of source tasks. After convergence, a relatively small data set is put into the training of the target task. Note that there are three training methods the same as the transfer learning.

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Settings

We selected several real-time network traffic trace datasets to evaluate our algorithms: WITS ISPDLSL-I, ISPDLSL II [8], Set09 and Set10 [9]. Some of the important features of the traces are listed in Table I. In this paper, the feature extraction is not a major topic of exploration. We select 16 features that promised unbiased feature selection based on previous studies [4]–[7].

We adopt the overall accuracy evaluate the performance of the models while considering the convergence time. When the accuracy does not increase more than one percent in the following 100 epochs, we consider the training as converged. And the Euclidean distance is adopted to measure the knowledge space. Considering a multi-dimensional mathematical space, d_{ij} denotes the normalized distances to the geometrical center of label j for instances with label i . The perplexity is defined as

$$perplexity = \frac{\sum_i^N d_{ii}}{\sum_i^N \sum_j^N d_{ij} - \sum_i^N d_{ii}}. \quad (6)$$

A lower perplexity indicates a better knowledge space representation of the traffic.

TABLE II
MULTI-OUTPUT DNN AND OTHER METHODS

Domain	Set	KNN	SVM	RF	DNN	Multi-output
Dure	I	92.70	94.64	96.32	96.08	96.09
	II	92.44	95.52	95.58	95.82	95.89
	09	86.86	88.16	83.54	94.90	95.15
	10	87.42	86.26	85.67	94.16	94.72
Rate	I	93.61	95.68	97.76	97.71	98.05
	II	92.46	95.95	95.88	98.17	97.81
	09	90.10	90.11	93.26	98.39	98.38
	10	90.48	88.02	91.19	97.77	98.38
App	I	96.36	96.13	96.50	97.09	96.54
	II	96.50	96.67	91.05	97.26	96.16
	09	97.96	96.80	85.67	98.36	98.20
	10	97.96	94.00	95.68	98.67	98.54

B. The Multi-Output DNN

In general, three tasks (duration, flow rate, and flow application classification) are discussed in the following discussion. There have been many papers on this topic and Murat Soysal et al. [7] has introduced a comprehensive comparison. Yet due to the difference of datasets and that most related research did not introduce much improvement on the adopted algorithms themselves, we choose to implement some standard machine learning algorithms which have been proved to perform well: the kNN, the SVM, and the RF (random forest). And we construct the DNN and the multi-output DNN with the same parameters and settings.

Table II illustrates the result of the overall accuracies for the three classification tasks with the above tools. Dure refers to the flow duration, and I, II, 09, and 10 correspond to dataset ISPD SL-I, ISPD SL-II, Entry09 and Entry10 respectively. We find that the DNN retains stable performances for all three tasks over different traces. It presents either the best or the second-best results in all the experiments. And multi-output DNN achieves approximately equally to the single DNN. Its performance shows the potential ability to handle multi-task problems and thus properly meet some of our proposals.

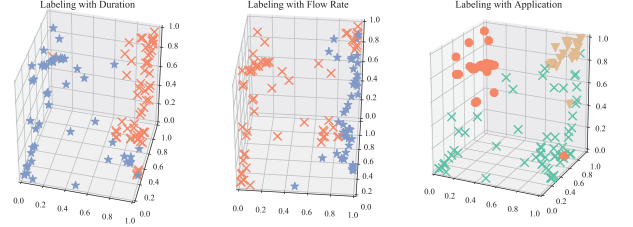
C. Common Knowledge

Table III demonstrates the perplexities of the different tasks represented by the common knowledge. It shows that the perplexities of the common knowledge extracted by the multi-output DNN are the lowest of all three tasks. Each of the single DNNs has the second-least perplexity for the corresponding training task but shares relatively higher ones for the other tasks. We find that the common knowledge extracted by our model represents the flows even better than the corresponding knowledge space of single task training.

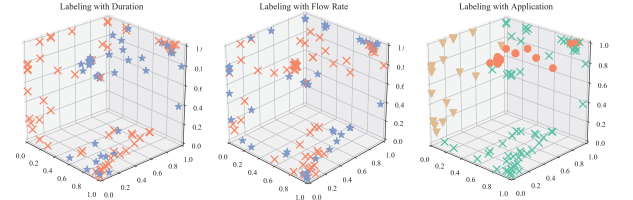
We go through the result spaces and select three neurons as the delegation. This is illustrated in the 3-dimensional graphs that are labeled with the three tasks in Fig. 2. Figure 2a shows that the values of the different labels are properly represented and separately clustered after some training epochs. In contrast, Fig. 2b demonstrates the knowledge space in a single-

TABLE III
PERPLEXITY METRICS OF THE COMMON KNOWLEDGE

Task \ Technique	Multi	Duration	Flow Rate	App
Duration	0.7440	0.7623	0.7677	0.9618
Flow Rate	0.7972	0.8486	0.8379	0.9603
Application	0.0672	0.0783	0.0783	0.0684



(a) The extracted common knowledge for different labeling.



(b) The "knowledge space" of single-task application.

Fig. 2. The knowledge space of multi-task and single-task training. Different marks and colors refer to different labels.

task training scenario, where the DNN properly extracts the knowledge labels with the application types. However, the knowledge space of DNN specified in one single task training does not fit the other two tasks.

D. Transfer Learning

In this section, we first train two tasks as source tasks. Then, the third task is raised as the target task and is concatenated to the established neural network. Three classification tasks are considered as the target task in turn. As there is no related research on transfer learning in traffic classification, we choose a single DNN, which shows the most outstanding performance among other methods with the same parameters with the multi-output DNN as the standard control. Fig. 3 illustrates the results of the three approaches and the standard control experiment.

The results show that the SCT and OCT share approximately equal accuracies with the single DNN while having fewer time costs over all the tests. This could attribute to the common knowledge extracted by source tasks. And it takes the least time to converge for OT in most cases, though its accuracy is not as good as others. Note that the multi-output DNN requires much less memory compared to constructing a new DNN while achieving competitive results.

E. Few-Shot Learning

Similar experiment setting to the last section is adopted. The difference is that we use only one percent of the total target

TABLE IV
FEW-SHOT LEARNING EXPERIMENTS

Source v.s. Target Task	Dataset	Single DNN		SCT		OCT		OT	
		Accuracy	Time, s	Accuracy	Time, s	Accuracy	Time, s	Accuracy	Time, s
flow rate, app v.s. duration	ISPDSL-I	87.44%	5.1	88.50%	1.8	90.75%	3.8	88.90%	3.9
	ISPDSL-II	90.03%	3.6	92.76%	2.7	92.07%	1.8	88.65%	0.9
	Entry09	85.73%	13.8	90.75%	5.0	92.22%	3.0	90.78%	2.5
	Entry10	75.62%	2.1	90.37%	1.5	90.40%	1.8	89.65%	1.4
duration, app v.s. flow rate	ISPDSL-I	87.61%	2.1	90.95%	2.0	91.21%	2.7	90.87%	1.6
	ISPDSL-II	91.39%	4.5	92.84%	4.1	93.56%	4.8	92.47%	3.0
	Entry09	79.92%	11.4	95.75%	6.1	96.02%	3.1	95.48%	1.5
	Entry10	79.83%	15.2	92.63%	3.4	94.92%	2.2	95.64%	3.5
duration, flow rate v.s. app	ISPDSL-I	86.81%	5.0	90.43%	7.5	89.16%	3.4	89.17%	2.9
	ISPDSL-II	90.17%	10.8	91.02%	10.5	90.49%	9.5	89.29%	3.1
	Entry09	95.21%	9.6	95.93%	12.2	95.86%	9.4	95.66%	10.6
	Entry10	89.29%	14.4	92.10%	5.1	93.76%	6.8	92.09%	5.4

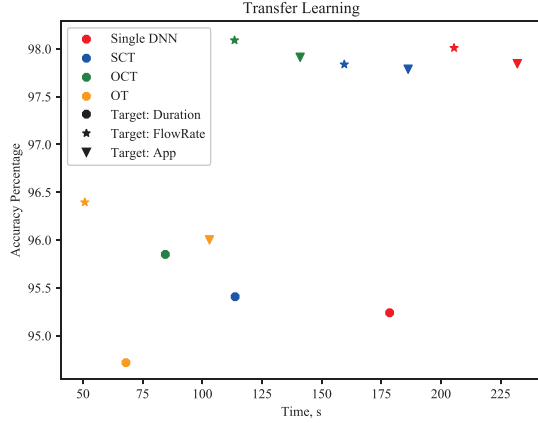


Fig. 3. Average accuracies and time costs for the transfer learning on different target tasks over all the datasets.

task data to be the input, and validate with the rest. Table IV shows the results of our few-shot experiment.

Under the condition of insufficient data, the DNN does not perform well in some of the cases. The accuracy achieves 70 to 80 percent in many cases. In contrast, the average accuracy of the multi-output DNN is approximately 94 percent regardless of the training scheme. In particular, it shows that the SCT and OCT have faster convergence speed and have higher accuracies than the standard control in most cases. Though accuracies of OT scheme are slightly lower than the other two schemes, it still advantages the single DNN on the accuracy. And it converges much faster than all the other schemes.

IV. CONCLUSION

In this paper, we discussed an approach to the traffic classification. First, we evaluated the effectiveness and the validation of the multi-output DNN for traffic classification tasks. Multi-output DNN is shown as a good approach to multi-task classification. Next, we explored the performance of common knowledge extraction. Experiments show that the structure can

properly extract the common knowledge of traffic, and the resulting knowledge space is even less perplexed than that of the corresponding single-task DNNs. We further evaluate the potential of our proposed algorithm to address new demands. All three training schemes have addressed the demands of the new classification of related tasks at a faster speed in our experiments. In addition, in few-shot learning scenes, the multi-output DNN shows remarkable results since it can be applied to achieve an even better accuracy result and a faster training speed with the support of common knowledge.

ACKNOWLEDGMENT

This work was jointly supported by: (1) National Natural Science Foundation of China (No. 61771068, 61671079, 61471063, 61372120, 61421061); (2) Beijing Municipal Natural Science Foundation (No.4182041, 4152039); (3) the National Basic Research Program of China (No. 2013CB329102).

REFERENCES

- [1] N. Feamster, J. Rexford, and E. W. Zegura, "The road to SDN: an intellectual history of programmable networks," *Computer Communication Review*, vol. 44, no. 2, pp. 87–98, 2014.
- [2] T. T. T. Nguyen and G. J. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1-4, pp. 56–76, 2008.
- [3] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Computer Networks*, vol. 132, pp. 81–98, 2018.
- [4] R. C. Jaiswal and S. D. Lokhande, "Machine learning based internet traffic recognition with statistical approach," in *India Conference (INDICON), 2013 Annual IEEE*. IEEE, 2013, pp. 1–6.
- [5] S. S. L. Pereira, J. L. de Castro e Silva, and J. E. B. Maia, "NTCS: A real time flow-based network traffic classification system," in *Proc. CNSM, Rio de Janeiro, Brazil, Nov. 2014*, pp. 368–371.
- [6] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1257–1270, 2015.
- [7] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Perform. Eval.*, vol. 67, no. 6, pp. 451–467, 2010.
- [8] "Waikato internet traffic storage," <https://wand.net.nz/wits/index.php>.
- [9] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proc. SIGMETRICS*, Banff, Alberta, Canada, Jun. 2005, pp. 50–60.