A survey of Knowledge Distillation

Trung Dao
VinAI Research
14 Thuy Khue, Tay Ho, Ha Noi
v.trungdt21@vinai.io

Abstract

For the last few years, deep learning has proved its ability to solve complicated problems that were too cumbersome to handle. It is undeniable that key factors contributing to this sucess are mainly large-scale data and deep neural networks with billions of parameters. However, to deploy and smoothly run these enormous models on on-edge devices is not always a straightforward process. In order to overcome this, a variety of model compression and acceleration techniques have been developed. As one of the outstand techniques, knowledge distillation (KD) seeks to transfer information learned from one large model (called teacher) to smaller one (called student). This survey firstly gives a brief overview of what KD is then quickly investigates some of the recent work about training schemes and teacher-student architectures.

1. Introduction

After achieving extraordinary establishment in various fields of Machine Learning using neural networks, scientists began to move interests into network compresssion and enhancement. Several worth-mentioned approaches which seek to create smaller model and cost-efficient are: parameter pruning/sharing, model quantization, low-rank factorization and knowledge distillation. A comprehensive overview on these approaches is outside of the scope of this survey, and the focus of this paper is solely about knowledge distillation, which has attracted many attentions from the deep learning research community in the recent years.

Comparing to other compression methods, KD might be considered superior since it can squeeze down a network disregarding the structural difference between the teacher and student network. The original idea of KD was proposed in [3], where the author focused on transfering the information from a large model or an ensemble of models into a smaller model without performance loss. This work is later generalized and popularized by Hinton et al. [9]. In their work, they found out an astounding observation that it is simplier to train a smaller-scale network (classifier) using the soften predictions of another classifier as target value rather than the ground truth (one-hot) labels and later on called this procedure as *distillation* (short for knowledge distillation). They also stated that these probabilities offer richer information than labels alone, and empirically help the student network learn better. Knowledge distillation field since then has been developed and applied in various forms, but the main characteristic of KD is still remained the same: Teacher-Student framework, where the teacher model provides useful *knowledge* to the student model in order to improve its learning performance.

Thanks to its effective on neural network compression and acceleration, KD has been widely applied in different fields of artificial intelligence: visual/speech recognition, natural language processing (NLP). In visual recognition, KD was firstly applied on classification tasks [9, 13, 4, 2, 5] then later on expanded to other applications such as image segmentation [7], lane detaction [10], facial landmark detection [20, 6]. In other field such as NLP, KD also proves its worth as being used to compress complex structures such as BERT [16, 12]. To briefly generalized, KD offers not only lightweight deep models, which allows deploying models to on-edge devices efficiently but recently also competive performant ones.

The main contributions of this paper consist of:

1. Provide a comprehensive overview of Knowledge Distillation: problem definition, type of knowledge, and its recent progress.
2. Investigate how research community try to theoretically explain KD.
3. Offer several scenarios and its feasible solution based on other recent works.

2. Background: Knowledge Distillation

2.1. Problem statement

Neural network models have been successful in a myriad of fields including extremely complex problems. However, these models are enormous in size and computational hungry, thus cannot be deployed to on-edge devices or simply not feasible in some situations. To compensate this situation, knowledge distillation was first proposed in [3] and properly formalize in [9]. Buciluǎ et al. [3], the knowlede is transferred from the larger model to smaller model while minimizing the logits difference produced by those two models respectively. This vanilla method open the first path of knowledge type that can be use to distilled, later on, there started to have some other works using the activations, feature maps of intermediate layers as the guide for the student network [14, 18, 8]. Some other methods to extract knowledge by comparing relationship between difference layers of teacher model [17, 19] are also worth mentioning but won't be discussed in this survey.

2.2. Knowledge

2.2.1 Knowledge from logits

In this type of knowledge extraction, it usually refers to using the neural response of the final output layer, or also known as *logits* of the teacher model. As stated before, the first application of logits in KD is used in [3], but in many situations, given a highly confident teacher model, the output of softmax function gives more less the same information as the ground truth label (since this is the core idea of softmax function - maximizing the probabilities of one while minimizing the others). Tackling this problem, Hinton et al. [9] proposed the concept of 'soft labels' and declared that this type of label contains informative *dark knowledge*. Given the **logits** z from a network, the 'soft label' p_i of an image is defined as:

$$p_i = \frac{\exp(\frac{z_i}{\rho})}{\sum_j \exp(\frac{z_i}{\rho})}$$

where ρ is the temperature parameter, notice that when $\rho = 1$, we get the normal softmax function. Determine which value is optimal for ρ is still a debatable topics but it is argued that while increasing ρ, the label becomes softer and providing more information about which class is similar to the predicted label. Accordingly, the logits-based distillation loss function is defined as follow:

$$\mathcal{L}_{\text{LB}}(x;\theta) = \alpha * \mathcal{H}(y, \sigma(z_s)) + \beta * \mathcal{H}(\sigma(z_t;\rho), \sigma(z_s;\rho))$$

where x is the input, θ are the student model weights, $\mathcal{H}(.)$ is the cross-entropy loss function, y is the ground truth label, $\sigma(.)$ is the ρ-parameterized softmax function, α, β are the coefficients to tune in order to balance between both cross-entropies, and z_t, z_s are the logits of teacher and student model respectively.

Logits-based knowledge is fairly straightforward and easy to implement so it is often the first approach used when need to apply knowledge distillation. As far as the survey goes, there are two main motives of using logits-based knowledge: 1) using soft labels 2) create/use noisy data to train. A brief table containing its description and several related work can be found at **??**. Nevertheless, since this type of knowledge only utilize the final output of the teacher model, it fails to offer information gathered in the intermediate layers, which later was proved to contain many informative result [14]. Other than that, due to its characteristic, logits-based knowledge is bounded with the supervised learning.

2.2.2 Knowledge from intermediate layers

What's extraordinary about deep neural networks is how they extract and abstractly represent features through the feature maps generated between each layer. Hence, feature-based knowledge would theoretically contain richer information than logits-based knowledge. Not only so, by combining both intermediate and last layer's output, we can consider feature-based knowledge is an extension of the previous knowledge type.

Romero et al. [14] introduced the term *hint* as the outputs of a teacher hidden layer, which are used to guide the student learning process. The core idea is to choose some intermediate layers in both teacher and student layer and force the student to mimics the result of the teacher's feature maps. Motivated by this work, there has been many studies to investigate which hint layer/ guided layer to choose and how to measure the distance between them. The feature-based distillation loss function is often defined as follow:

$$\mathcal{L}_{FB}(f_t, f_s) = \mathcal{D}(\Phi_t(f_t), \Phi_s(f_s))$$

where f_t, f_s are the selected hint and guided layers, Φ_t, Φ_s are the transformation functions for the mentioned layers respectively, which often are applied when the result of two layers are mismatched, $\mathcal{D}$ is the distance function measuring the similarity between the hint and guided layer. L1, L2 is often used for this distance function, but for some works, more complex function might be used such as the Maximum Mean Discrepancy (MMD) metric [11] or Kullback–Leibler divergence [1].

Eventhough this knowledge type offers more favorable information for the student learning process, where to pick the hint/ guided layers or the transformation functions still need more investigations. For examples, the teacher's layer transformation function are directly critical for this process since there are risks of losing information in the process of transforming. Several past work has already encountered this problem since the feature map's dimension got downscaled, causing loss of knowledge [18, 15]. To counter this situation, one could either come up with novel transformation function such as margin RELU[8] or not use any transformation at all [14]. But as investigated by Heo et al. [8], the hints consist of both beneficial and adverse information so we should try to only gather the positive ones rather than distilling all of it. The same goes for the guided layers, sometimes researchers use the same transformation [18], which might lead to the same information loss. Some other works use 1×1 convolutional layer as a student transform [14, 8] in order to match the depth dimension with the hints, so no information would be lossed during the process. A more detail of progressive work using feature-based knowledge can be found at figure **??**.

3. Explaining Knowledge Distillation

As discussed in the previous sections, knowledge distillation has been successfully applied in various fields and applications. However, clarying how and why KD works in general and its models are usually superior than those trained from raw data still remains an work in progress.

References

[1] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations, 2020.

[2] Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi. Label refinery: Improving imagenet classification through label progression, 2018.

[3] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535–541, New York, NY, USA, 2006. Association for Computing Machinery.

[4] Wei-Chun Chen, Chia-Che Chang, Chien-Yu Lu, and Che-Rung Lee. Knowledge distillation with feature maps for image classification, 2018.

[5] Joseph DiPalma, Arief A. Suriawinata, Laura J. Tafe, Lorenzo Torresani, and Saeed Hassanpour. Resolution-based distillation for efficient histology image classification, 2021.

[6] Shiming Ge, Shengwei Zhao, Chenyu Li, and Jia Li. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Transactions on Image Processing*, 28(4):2051–2062, Apr 2019.

[7] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge adaptation for efficient semantic segmentation, 2019.

[8] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation, 2019.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[10] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation, 2019.

[11] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer, 2017.

[12] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2020.

[13] Zhizhong Li and Derek Hoiem. Learning without forgetting, 2017.

[14] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015.

[15] Zhiqiang Shen, Zhankui He, and Xiangyang Xue. Meal: Multi-model ensemble via adversarial learning, 2019.

[16] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[17] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7130–7138, 2017.

[18] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, 2017.

[19] Chenrui Zhang and Yuxin Peng. Better and faster: Knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification, 2018.

[20] Yang Zhao, Yifan Liu, Chunhua Shen, Yongsheng Gao, and Shengwu Xiong. Mobilefan: Transferring deep hidden representation for face alignment. *Pattern Recognition*, 100:107114, 2020.