MULTILAYER PROBABILISTIC KNOWLEDGE TRANSFER FOR LEARNING IMAGE REPRESENTATIONS

Nikolaos Passalis, Maria Tzelepi and Anastasios Tefas

Dept. of Informatics, Aristotle University of Thessaloniki, Greece
Email: {passalis, tzelepi, tefas}@csd.auth.gr

ABSTRACT

Probabilistic Knowledge Transfer (PKT) aims to transfer the knowledge encoded in the representations extracted from a layer of a large and complex neural network (teacher) into a smaller and faster one (student). However, PKT only transfers the knowledge between two layers of the networks, ignoring the potentially useful information encoded by the previous ones, reducing in this way the efficiency of PKT and the performance of the student model. In this paper, we propose a novel efficient multilayer PKT method that is capable of transferring the knowledge between the student and teacher networks by employing the representations extracted from multiple layers. The ability of the proposed multilayer PKT method to improve the knowledge transfer and increase the performance of the student model over other state-of-the-art methods is demonstrated using two image datasets.

1. INTRODUCTION

The success of Deep Learning (DL), along with the increasing need to deploy DL models on embedded and mobile devices, led to the development of a wide range of methods for training faster and smaller DL models, which have lower energy and computational footprint. Several methods have been proposed to this end, including model compression and quantization approaches [1], pruning methods [2], lightweight DL models [3], as well as knowledge transfer/distillation [4] and regularization methods [5, 6]. Quantization methods focus on lowering the number of bits needed for storing the weights of a network, pruning methods on discarding less important weights/neurons, lightweight model design approaches on creating more efficient DL model architectures, while knowledge transfer on improving the efficiency of the training process for smaller models. Knowledge transfer methods work by transferring the knowledge from a larger and more complex neural network, called *teacher*, to a smaller and faster one, called *student*. Knowledge transfer methods attracted significant research attention, since they can improve the performance of any DL model and can be easily combined with any other approach for developing more accurate lightweight DL models.

The vast majority of knowledge transfer methods aim at training more efficient models for classification tasks by transferring the knowledge between the last (classification) layers of the networks [4]. However, the vast majority of these methods cannot be used efficiently for representation learning tasks, where other activation functions than the softmax are used and the dimensionality of the layers between the networks is different, leading to the development of methods specifically designed to handle such tasks [7, 8]. Among the most powerful approaches for handling representation learning tasks is the Probablistic Knowledge Transfer (PKT) method, which can overcome the aforementioned limitations and transfer the knowledge encoded in the representations extracted from any layer into a lightweight DL model. PKT can indeed handle several different knowledge transfer scenarios, ranging from cross-domain knowledge transfer to transferring the knowledge from handcrafted feature extractors.

However, PKT only transfers the knowledge between two specific layers, ignoring the knowledge that is contained in earlier layers of the teacher model. In this way, PKT currently ignores the potentially useful information encoded in the earlier layers, reducing the efficiency of the knolwedge transfer process. The main difficulty for using PKT for multilayer knowledge transfer arises when networks with vastly different architectures are used, e.g., when the number of layers is different between the student and teacher. In these cases, it is not straightforward to select the intermediate layers which should be used for transferring the knowledge. For example, suppose that we are transferring the knowledge from a 6-layer teacher into a 4-layer student. It is not clear if we should use the first or second layer of the teacher for transferring the knowledge to the first layer of the student. Selecting the most appropriate layer is especially important, since selecting the wrong layer for transferring the knowledge can have a devastating effect on the accuracy of the network, either by over-regularizing the network [9], or by reducing the granularity of information analysis, leading to worse performance compared to not using multilayer transfer. Therefore, even though PKT can support knowledge transfer between any two layers of two neural net-

This project is implemented through the Operational Program ”Human Resources Development, Education and Lifelong Learning” and is co-financed by the European Union (European Social Fund) and Greek national funds (MIS Code: 5047925).

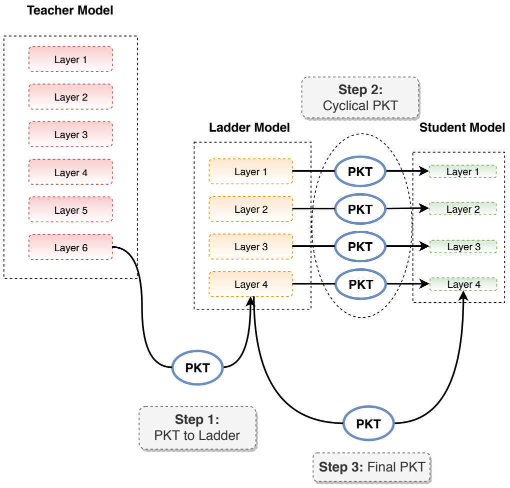

Fig. 1. Multilevel PKT: Performing the proposed 3-step process for transferring the knowledge between multiple layers of the teacher and student models

works, there is currently no efficient way of selecting the most appropriate layers to use for this task.

The main contribution of this paper is a multilayer knowledge transfer approach, that allows for effectively using PKT to exploit the knowledge encoded in the intermediate layers of the teacher model. The proposed method works by first transferring the knowledge into an intermediate network, called *ladder* model, that acts as a proxy to the teacher model. The ladder network is smaller than the teacher model, but larger than the student model, and it is designed to have compatible architecture with the student model. In this way, the layers between the ladder and student models can be directly matched, as shown in Fig. 1. Then, the knowledge is transferred from the ladder model to the teacher model using a cyclical training procedure, that allows for better exploring the solution space. To the best of our knowledge, this is the first time that it is demonstrated that multilayer probabilistic knowledge transfer can be effectively performed using vastly different neural networks architectures. Finally, the effectiveness of the proposed method is experimentally demonstrated using two different datasets and knowledge transfer setups.

The rest of the paper is structured as follows. In Section 2 the related work is briefly introduced and discussed, while the proposed method is presented in Section 3. Then, the experimental evaluation is provided in Section 4 and conclusions are drawn in Section 5.

2. RELATED WORK

A large number of knowledge transfer methods which build upon the neural network distillation approach have been proposed [4, 10, 11, 12]. These methods employ the teacher model to generate soft-labels and then use these soft-labels for training the smaller student network. Several extensions to this approach have also been proposed. For example, soft-labels can be used for pre-training a large network [13] and performing domain adaption [12], while an embedding-based approach for transferring the knowledge was proposed in [14]. Furthermore, knowledge transfer methods have been recently extended to handle representation learning tasks [7, 8]. However, these methods only focus on transferring the knowledge between the classification layers of the networks and ignore the knowledge encoded in the earlier layers of the networks. Also, using a proxy network for improving knowledge transfer was proposed in [15]. However in contrast with the proposed method, the proxy network used in [15] was employed to merely improve the performance of knowledge transfer between two layers, instead of designing a proxy that can facilitate efficient multilevel knowledge transfer, as proposed in this paper.

In contrast with the aforementioned approaches, the proposed method provides a way to perform multilevel knowledge transfer, exploiting the knowledge encoded by the earlier layers of a neural network. It is also worth noting, that existing methods that support multilayer knowledge transfer, such as using hints [9], or the flow of solution procedure matrix (FSP) [16], usually only target networks with compatible architecture, e.g., residual networks with same number of residual blocks, for both the teacher and student. However, the proposed method provides an efficient way for handling any possible network architecture by employing a ladder proxy. To the best of our knowledge, in this work we propose the first probabilistic knowledge transfer approach that can effectively exploit the knowledge encoded in various levels of the teacher network to further improve the student model.

3. PROPOSED METHOD

Let $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N\}$ denote the *transfer* set, composed of N images used to transfer the knowledge encoded in the teacher model into the student model. Also, let $\mathbf{x}^{(l)} = f(\mathbf{t}, l)$ denote the representation extracted from the l-th layer of the teacher model $f(\cdot)$ and $\mathbf{y}^{(l)} = g(\mathbf{t}, l, \mathbf{W})$ denote the representation extracted from the l-the layer of the student model $g(\cdot)$. Note that the trainable parameters of the student model are denoted by $\mathbf{W}$. PKT aims to train the student model $g(\cdot)$ in order to “mimic” the behavior of $f(\cdot)$. In [7], it is demonstrated that minimizing the divergence between the teacher's and student's conditional probability distributions, which are estimated as:

$$p_{i|j}^{(t,l)} = \frac{K(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)})}{\sum_{i=1,i\neq j}^{N} K(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)})} \in [0, 1], \qquad (1)$$

and

$$p_{i|j}^{(s,l)} = \frac{K(\mathbf{y}_i^{(l)}, \mathbf{y}_j^{(l)})}{\sum_{i=1,i\neq j}^{N} K(\mathbf{y}_i^{(l)}, \mathbf{y}_j^{(l)})} \in [0, 1], \qquad (2)$$

where $K(\cdot)$ is a kernel function, provides an effective way for transferring the knowledge encoded in the teacher into the student. Note that we assume, without loss of generality, that for both networks, the l-th layer is employed for the knolwedge transfer. These probabilities express how probable is for each sample to select each of its neighbors [17], modeling in this way the geometry of the feature space, while matching these two distributions also ensures that the mutual information between the models and a set of (possibly unknown) classes is maintained [7].

Note that the kernel choice can have a significant effect on the quality of the knowledge transfer. Apart from the well known Gaussian kernel, which is however often hard to tune, other kernel choices include cosine-based kernels, e.g., $K_c(\mathbf{a}, \mathbf{b}) = \frac{1}{2}(\frac{\mathbf{a}^T\mathbf{b}}{||\mathbf{a}||_2||\mathbf{b}||_2} + 1)$, and the T-student kernel, i.e., $K_T(\mathbf{a}, \mathbf{b}) = \frac{1}{1+||\mathbf{a}-\mathbf{b}||_2^d}$, where d is typically set to 1. Selecting the most appropriate kernel for the task at hand can lead to significant performance improvements, e.g., cosine-based kernels perform better for retrieval tasks, while using kernel ensembles, i.e., estimating the probability distribution using multiple kernels, can also improve the robustness of PKT. Therefore, in this paper a hybrid objective that aims at minimizing the divergence calculated using both the cosine kernel, which ensures the good performance of the learned representation for retrieval tasks, and the T-student kernel, which ensures the good performance of the method for classification tasks is used: $\mathcal{L} = \mathcal{D}(\mathcal{P}_c^{(t,l)}||\mathcal{P}_c^{(s,l)}) + \mathcal{D}(\mathcal{P}_T^{(t,l)}||\mathcal{P}_T^{(s,l)})$, where $D(\cdot)$ is a divergence metric and the notation $\mathcal{P}_c^{(t,l)}$ and $\mathcal{P}_T^{(t,l)}$ is used to denote the conditional probabilities of the teacher calculated using the cosine and T-student kernels respectively. Again, we assume that the representations used for knowledge transfer were extracted from the l-th layer. The student probability distribution is denoted similarly by $\mathcal{P}_c^{(s,l)}$ and $\mathcal{P}_T^{(s,l)}$. The divergence is calculated using a symmetric version of the Kullback-Leibler (KL) divergence, the Jeffreys divergence: $\mathcal{D}(\mathcal{P}^{(t,l)}||\mathcal{P}^{(s,l)}) = \sum_{i=1}^{N}\sum_{j=1,i\neq j}^{N}\left(p_{j|i}^{(t,l)} - p_{j|i}^{(s,l)}\right) \cdot \left(\log p_{j|i}^{(t,l)} - \log p_{j|i}^{(s,l)}\right)$, which can be sampled at a finite number of points during the optimization, e.g., using batches of 64-128 samples. Finally, stochastic gradient descent is employed to train the student model: $\Delta\mathbf{W} = -\eta\frac{\partial\mathcal{L}}{\partial\mathbf{W}}$, where $\mathbf{W}$ is the matrix with the parameters of the student model and η is the employed learning rate.

Using multiple layers to transfer the knowledge is expected to better guide the knowledge transfer process. This has been also demonstrated for classification tasks, using hints from multiple layers in [9]. However, using hints requires carefully selecting the layers that will be used for the knowledge transfer to avoid over-regularizing the student. Indeed, if the intermediate layers are not carefully selected, the performance of the student model is often worse than not using multilayer transfer at all. Unfortunately, due to the poor understanding of the way that neural networks transform the probability distribution of the input data, there is currently no way to select the most appropriate layers for transferring the knowledge *a priori*, without manually evaluating various combinations of layers. This process can be especially difficult and tedious, especially when the architectures of the student and teacher differ a lot. In this work, we proposed to overcome this limitation by constructing an appropriate proxy for the teacher model, that will allow for directly matching between all the layers of the proxy model and the student model, as shown in Fig. 1. In this way, the proposed method employs the intermediate proxy, called *ladder* network, that has compatible architecture with the student model, to better facilitate the process of knowledge transfer.

The proposed method works as follows: First, the knowledge is transferred from the teacher model to the ladder model using only the final representation layers of the networks (step 1 in Fig. 1), as originally proposed in [7]. The ladder model is designed to be stronger than the student model, yet to have compatible architecture. Thus, the ladder model is expected to perform better compared to the student. Then, all the layers of the student and ladder networks are used to transfer the knowledge, without the risk of mismatching between the layers. We propose to design the ladder network using the same architecture as the student model, but using more neurons/convolutional filters per layer. Thus, the greater learning capacity of the ladder network ensures that enough knowledge will be always available to the ladder network (when compared to the student model), leading to better results compared to directly transferring the knowledge from the teacher model.

Then, the knowledge is transferred between all the layers of the student and ladder networks using a cyclical training process (step 2 in Fig. 1). More specifically, instead of transferring the knowledge from all the layers simultaneously, which requires fine-tuning the weight for the loss induced by each layer, we propose randomly selecting a pair of two layers and performing one full transfer epoch using the specific layer pair. Then, this process continues by selecting another layer pair, until completing a predefined number of training epochs. We found out that this process allows for a) more easily using the proposed method without having to select any hyper-parameters, as well as b) better exploring the solution space. Finally, the teacher model is fine-tuned for a number of training epochs (step 3 in Fig. 1) using regular PKT between the final representation layers.

4. EXPERIMENTAL EVALUATION

The proposed method was evaluated using the CIFAR-10 [18] and STL-10 [19] datasets. For all the conducted experiments, the teacher network was a ResNet-18 network [20] trained for classification using the CIFAR-10 dataset. The penultimate layer of the teacher network was used to transfer the knowledge. The ladder network was composed of a 3 × 3 convolutional layer with 16 filters, followed by a 2 × 2 max pooling layer, another 3 × 3 convolutional layer with 32 filters, a 2 × 2 max pooling layer, a 3 × 3 convolutional layer with 64 filters, another 2 × 2 max pooling layer and a final fully connected layer with 128 neurons. The ReLU activation function was used for all the layers, while batch normalization was also employed for the convolutional layers. The student model has the same architecture (number of layers) as the ladder, but uses half the number of filters/neurons at each layer. The teacher was trained for 50 epochs with a learning rate of 0.001, followed by an additional 30 training epochs with a lowered learning rate of 0.0001. For all the conducted experiments, the Adam algorithm [21] was employed for the optimization, while the batch size was set to 128. Finally, the optimization for the proposed method ran for 50 epochs (20 for the STL-10 dataset).

The proposed method, abbreviated as "M-PKT", was compared to four other knowledge transfer approaches: a) hint-based transfer (where the projection matrix was optimized during the knowledge transfer process) [9], b) distillation transfer (using an additional classification layer) [4], c) MDS-based transfer, as proposed in [8], and d) regular PKT [7]. The retrieval evaluation setup proposed in [7] was used, while the mean average precision (mAP) and top-50 precision (t-50) using both the euclidean similarity metric "(e)" and the cosine similarity "(c)" are reported. The retrieval metrics are also provided for the teacher, ladder and student models trained directly for classification tasks, while the student model was initialized using the pre-trained student. The optimization ran for 50 training epochs with a learning of 0.001 and 20 training epochs with a learning rate of 0.0001 for all the evaluated methods. Global average pooling is employed for the feature maps extracted from the intermediate convolutional layers, when used to estimate the probability distributions for the proposed M-PKT method.

The evaluation results for the CIFAR-10 dataset are reported in Table 1. Note that the hint-based transfer and the PKT method lead to consistent and significant improvements over directly training the student using the ground truth labels provided by the CIFAR-10 dataset. Also, note that despite the vastly different architectures between the ResNet-18 teacher and the employed student model, the proposed multilevel transfer method was capable of improving the mAP over the plain PKT method by more than 1.5% (relative increase).

The proposed method was also evaluated in a more challenging distribution shift setup, where the STL-10 dataset was used for transferring the knowledge and evaluating the performance of the methods. The images from STL-10 dataset were resized to 32 × 32 pixels, in order to be compatible with the networks trained on CIFAR-10. The same teacher, ladder and student models are used as before. The optimization ran for 20 training epochs with a learning of 0.001 and 10 training epochs with a learning rate of 0.0001. The experimental results are reported in Table 2. Using any of the knowledge transfer methods leads to significant improvements over directly training the student, while the proposed method still outperforms all the other evaluated methods. Note that the proposed method leads to even larger improvements compared to the previous experiments, e.g., the mAP increases by more than 4% (relative increase) over the plain PKT method, confirming the effectiveness of the proposed method.

Table 1. CIFAR-10: Retrieval evaluation

Method	**mAP (e)**	**mAP (c)**	**t-50 (e)**	**t-50 (c)**
Teacher	87.18	90.47	92.40	92.45
Ladder	62.12	66.78	75.02	76.97
Student	29.15	31.79	47.75	49.93
Hint (optim.) [9]	32.28	36.71	50.41	52.67
Distillation [4]	28.44	31.26	48.17	50.47
MDS-T [8]	30.36	31.99	45.12	46.76
PKT [7]	37.05	40.09	50.28	52.96
M-PKT	**37.72**	**40.59**	**50.69**	**53.27**

Table 2. STL-10: Retrieval evaluation (distribution shift)

Method	**mAP (e)**	**mAP (c)**	**t-50 (e)**	**t-50 (c)**
Teacher	57.40	61.20	68.87	71.37
Ladder Teacher	45.61	49.63	57.57	60.77
Student	26.04	28.03	36.55	38.31
Hint [9]	30.31	33.58	41.26	43.68
Distillation [4]	28.11	30.35	39.77	41.66
MDS-T [8]	28.85	30.91	37.51	39.80
PKT [7]	30.68	32.71	39.61	41.72
M-PKT	**31.92**	**34.15**	**41.11**	**43.61**

5. CONCLUSIONS

A novel multilayer knolwedge transfer method was proposed in this paper. The proposed method was capable of efficiently transferring the knowledge between the student and teacher networks by employing the representations extracted from multiple layers. To avoid the need for carefully matching between the layers of the student and teacher models, the proposed method employed an intermediate network, called ladder model, that acts as a proxy to the teacher model, significantly simplifying the knowledge transfer process. The effectiveness of the proposed multilayer probabilistic knolwedge transfer method was demonstrated using experiments on two image datasets. Several interesting research questions arise as a result of this study. Is there any other more structured way of designing and training auxiliary models that will further increase the performance of the student model? Furthermore, this paper focused on transferring the knowledge between convolutional neural networks. However, the proposed method can be also readily applied for transferring the knowledge between vastly heterogeneous architectures, such as recurrent neural networks or handcrafted feature extractors. Is it possible to use multiple and heterogeneous ensembles of auxiliary models to further improve the knowledge transfer?

6. REFERENCES

[1] Song Han, Huizi Mao, and William J Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint: 1510.00149*, 2015.

[2] Yihui He, Xiangyu Zhang, and Jian Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1389–1397.

[3] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint: 1704.04861*, 2017.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," in *Proceedings of the Neural Information Processing Systems Deep Learning Workshop*, 2014.

[5] Maria Tzelepi and Anastasios Tefas, "Class-specific discriminant regularization in real-time deep cnn models for binary classification problems," *Neural Processing Letters*, pp. 1–17.

[6] Maria Tzelepi and Anastasios Tefas, "Improving the performance of lightweight cnn models using minimum enclosing ball regularization," in *Proceedings of the European Signal Processing Conference*, 2019, pp. 1–5.

[7] Nikolaos Passalis and Anastasios Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 268–284.

[8] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2907–2916.

[9] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio, "Fitnets: Hints for thin deep nets," in *Proceedings of the International Conference on Learning Representations*, 2015.

[10] Zhiyuan Tang, Dong Wang, and Zhiyong Zhang, "Recurrent neural network training with dark knowledge transfer," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5900–5904.

[11] Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil, "Model compression," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2006, pp. 535–541.

[12] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[13] Zhiyuan Tang, Dong Wang, Yiqiao Pan, and Zhiyong Zhang, "Knowledge transfer pre-training," *arXiv preprint: 1506.02256*, 2015.

[14] Nikolaos Passalis and Anastasios Tefas, "Unsupervised knowledge transfer using similarity embeddings," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 946–950, 2018.

[15] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh, "Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher," *arXiv preprint: 1902.03393*, 2019.

[16] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7130–7138.

[17] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[18] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," *Technical Report*, 2009.

[19] Adam Coates, Andrew Ng, and Honglak Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[21] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.