Knowledge Distillation: A Free-teacher Framework Driven by Word-Vector

Chuyi Zou[1,a]

[1]*Nanchang Hangkong University*
Nanchang, China
a. 407217856@qq.com

***Keywords*：** Knowledge Distillation; Classification; Deep Learning.

***Abstract*：** Knowledge distillation (KD) is an effective method to transferring knowledge from a larger teacher network to a small student network, in order to enhance the generalization ability of the small student network, which satisfies the low-memory and fast running requirements in practice. Existing KD methods often require a pre-trained teacher as a first step to discover useful knowledge, then subsequently transferring knowledge to student network. However, this procedure is a two training complex stages, requiring an expensive computational cost for a pre-trained teacher. In this paper, we propose a free-teacher framework driven by word-vector to address this limitation. By utilizing existing word vector packets (such as 'GoogleNews-vectors-negative300', etc.), we are committed to create a semantic similarity matrix. This matrix provides the additional soft label which is similar to conventional teacher model’s outputs, while does not require any extra training cost. Extensive evaluations show that our approach improve the generalization performance of a variety of deep neural networks competitive to alternative methods on two image classification datasets: CIFAR10 and CIFAR100, whilst not requiring extra expensive training cost.

1. Introduction

Deep learning techniques[1] have gained massive success in a wide variety of vision problems, such as image classification[2-5], object detection[6], semantic segmentation[7], activity recognition[8], image captioning[9], and so forth. However, the supper performance of these deep network[10-12] always accompany with large model and a huge number of parameters, which limited their application when comes to computation resource limited environment. Recently, Hinton et al.[13] exploit the concept of knowledge distillation(KD) and show the effectiveness to transferring knowledge from a larger teacher network to a small student network, which suits for resource limited deployment. Specifically, they learn a heavy (e.g. higher-capacity) teacher model in a computationally intensive manner as the first step. Then extract the learned knowledge (i.e. inter-class correlations), which is the class probabilities given a sample, from the teacher model. Lastly optimize the student model by leveraging both the label data and softened output of teacher model (referred to as soft label). Via this procedure, the student network's performance achieves a large improvement. Based on this, Romero[14] find not only transfer the final output but also intermediate hidden layer values of the teacher network can make the student network even achieve higher performance than teacher network. At the same time, Zagoruyko[15] show that by properly defining attention for convolutional neural networks, will significantly improve the performance of a student CNN network by forcing it to mimic the attention maps of a powerful teacher network. However, existing KD approaches comes with extra high training costs in computation due to the need for additionally learning a heavy larger teacher model (an ensemble of networks or a higher capacity network) in order to obtain an additional knowledge source (e.g. inter-class correlations) for helping train the small (student) model. In real applications, this additional burden means higher power consumption and therefore is practically inferior in terms of economy and environment. In this paper, we propose a Free-teacher Framework Driven by Word-Vector (FFDWV) to address the aforementioned limitations. Instead of using the teacher model's soften output as the soft label to guide the student network, we utilize the existing word vector packets (such as 'GoogleNews-vectors-negative300'[16], etc.) to produce soft label yielding no extra cost for training a teacher model which resource limited environment are desperate for. Using word vector to formulate the soft label is inspired by the recent progress of natural language processing systems[17-19]. There are many packages of word vectors. For example, in the word-vector package (Google News, etc.), it quantifies most words in natural language into a number matrix that can be calculated by computer. Each word has a distance from each other. So in CIFAR10 and CIFAR100, these classes of the processed image classification task, such as dog and cat, can also be quantified among the corresponding words in the word-vector package. The centre of this paper is obtained, which directly uses the distance between the corresponding tags in the classification task in the word vector package, and then transforms it into the similarity matrix between the tags as the soft target, completely divorced from the teacher model.

This work makes the following contributions:

- We propose a generic Free-teacher Framework Driven by Word-Vector for knowledge distillation approach without requiring any additional computational cost to train a teacher.
  To best of our knowledge, this is first attempt to consider the efficiency of the knowledge distillation and provide a completely free teacher without any extra cost.

- We summarise the KD methods and compare them in a rigorous manner, in particular attempting to consider the efficiency of the KD. Extensive experiments demonstrate the superiority of our method compared to KD and vanilla method on CIFAR10 and CIFAR100.

2. Related Work

The key idea behind knowledge distillation have been proposed around a decade. Bucilua et al.[20] firstly propose an algorithm comprising the information in ensemble models into single model by force the single model to mimick the output of ensemble models. Ba and Caurana[21] extend this approach to model compression by learning on logits rather than the probability distribution. Based on Ba and Caurana[21], Hinton et al. propose “knowledge distillation” by introducing the tempreture before performing a softmax function in an ensemble of models (teacher model) to produce the ’soft label’. Through mimicing the soft label of teacher model, the student network's performance achieves a large improvement, while it requires a more training cost to training the teacher model. Besides, due to the student network is shallow than teacher network, the performance of student network still lower than teacher network. Based on their work, Romero et al[14] extend this idea to allow the training of a student that is deeper and thinner than the teacher network. They not only transfer the softened out of teacher network, but also the intermediate representations learned by the teacher as hints to improve the training process and final performance of the student network. Due to these intermediate representations transfer, the student network with fewer parameters even perform better than teacher network on several data sets. Wojciech et al[22] proposes sobolev training for neural network.

They not only approximate the teacher network's outputs but also using the teacher network's derivatives as encode additional information to train the student network.

However, existing KD approaches need two-stage sequential optimisation, and require a large computational cost to get a pre-trained large teacher model. In this study, we propose a Free-teacher Framework Driven by Word-Vector (FFDWV) to address this drawback. Utilizing the existing word vector packets (such as Google news word-vector packets, etc.), we are capable to produce semantic matrix which guides the student network optimization procedure.

3. A Free-Teacher Framework Driven By Word-Vector

3.1. Revisiting Knowledge Distillation

We begin with revisiting the conventional Knowledge Distillation proposed by Hinton [13]. Suppose we have accessed to n labelled training samples $D=\{(x_i,y_i)\}_i^n$, each of them belongs to category C , the large teacher network θ outputs a probabilistic class posterior $p(c\,|\,x,\theta)$ for a sample x over a class c as:

$$p(c\,|\,x,\theta)=f(z)=\frac{\exp(z^c)}{\sum_{j=1}^{C}\exp(z^j)}$$

□ □ □ □ □ □ □ (1)

Where z is the logits produced by the network θ. When training the model, we usually use the Cross-Entropy (CE) loss between the predicted value and the ground truth label as the objective loss function:

$$L_{ce}=-\sum_{c=1}^{C}\delta_c\log\left(p(c\,|\,x,\theta)\right)$$

□ □ □ □ □ □ (2)

Where δ_c is Dirac delta which returns 1 if c is the ground-truth label, and 0 otherwise. With the CE loss, we firstly train the teacher network in an end to end way.

After obtaining a pre-trained large teacher network, we able to transfer the knowledge from the teacher to student network by force the student network's output mimic the teacher's network output. We firstly extract the knowledge of the teacher network via its soft prediction given a sample:

$$\tilde{p}(c\,|\,x,\theta)=\frac{\exp(z^c/T)}{\sum_{j=1}^{C}\exp(z^j/T)}$$

□ □ □ □ (3)

In order to transfer knowledge from teacher to student, we force the output of student network mimic the output of teacher network through KL divergency:

$$L_{kl}=\sum_{j=1}^{C}\tilde{p}(j\,|\,x,\theta)\log\frac{\tilde{p}(j\,|\,x,\theta)}{p(j\,|\,x,\theta)}$$ □ □ □ □ □ □ (4)

Then, the total loss to optimize the student network is calculated by:

$$L=L_{ce}+T^{2}*L_{kl}$$ □ □ □ □ □ (5)

3.2. Free-teacher Framework Driven by Word-Vector

An overview of the Free-teacher Framework Driven by Word-Vector (FFDWV) architecture is depicted in Figure 1. The FFDWV consists of two process: 1) semantic similarity matrix production 2) knowledge transfer.

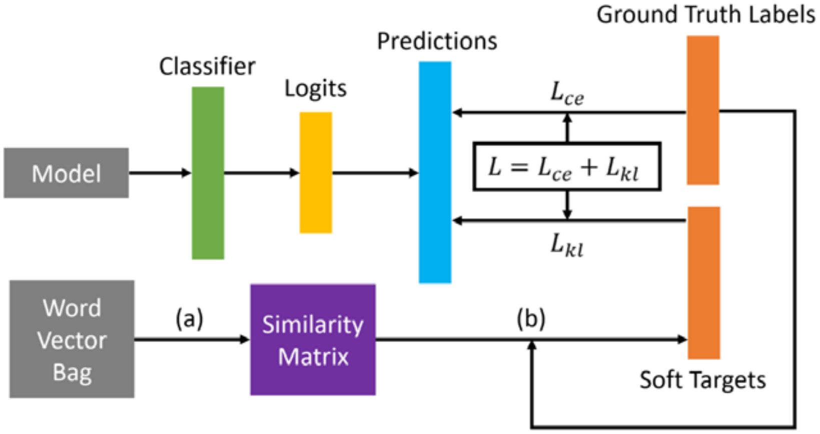

Figure 1: FFDWV method structure. (a) the word vector package ('GoogleNews-vectors-negative300') extracted from Google News can generate 10 * 10 and 100 * 100 matrices by listing the words of all classes in CIFAR 10 and CIFAR 100 (such as dog, cat). This matrix is the mutual distance between each class. Here we use Euclidean distance as the similarity between two words, and calculate the Euclidean distance from the generated distance matrix to get the similarity matrix. (b) find the row corresponding to the similarity matrix in the tag and use it as the soft target.

3.2.1 Semantic Similarity Matrix Production:

We use word vector package to generate the semantic similarity matrix among classes for specific classification task e.g. CIFAR10. We firstly obtain a set of word vector $w_i\in(w_0,w_1,\cdots,w_9)$. Where is the 10 classes number of CIFAR10. Then we calculate their distance according to Euclidean distance in two dimensional space:

$$w_{ij}=\sqrt{w_i^2+w_j^2}$$ □ □ □ □ □ □ (6)

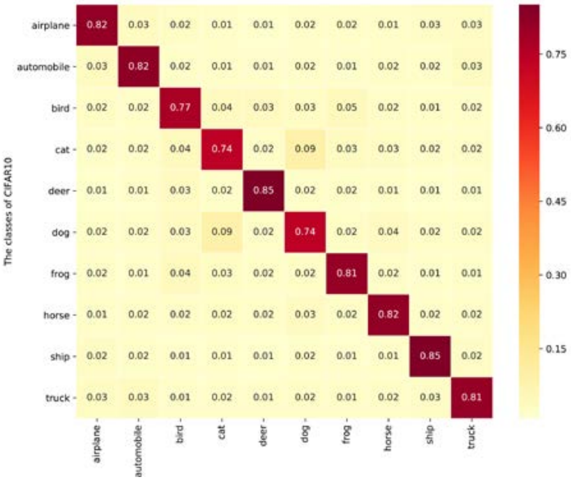

Figure 2: Word vector similarity matrix with CIFAR10 as an exampe (after softmax normalization).

Where w_{ij} is the distance between any two categories. The smaller their value is, the more similar the two categories are. We use softmax to normalize the distance matrix W and convert it to soft target:

$$s_j = \frac{\exp(-w_{ij})}{\sum_{i=0}^{9} \exp(-w_{ij})}, j \in \{0,1, \quad \cdots, 9\} \tag{7}$$

3.2.2 Knowledge Transfer

Through the Semantic similarity matrix production, we are able to obtain a set of soft label without the need of a pre-trained large teacher model, which will spend expensive training cost. In order to optimize the student model, we adopt the similar strategy with conventional KD method for FFDWV approach. For a given samples belong to j^{th} category, we use the j^{th} column of matrix S as its soft label s_j. Then, we train our model with cross-entropy[23] and the alignment loss with soft label s_j as follows:

$$L_{soft} = \sum_{j=0}^{C} s_j \log \frac{s_j}{p(j \mid x, \theta)} \tag{8}$$

The total is calculated as:

$$L = L_{ce} + L_{soft} \qquad □ □ □ \tag{9}$$

The overall process is in Table 1.

Remarks: Our FFDWV is generic free-teacher framework for knowledge distillation without increase extra any training cost. This reduce the complexity of two stage training produce of conventional KD methods. We do not add the temperature parameters T during the knowledge transfer, since our soft label produced by Word vector already obtain the soften prediction as Figure 2 show.

Table 1: Free-teacher Framework Driven by Word-Vector.

```
**Input**: Labelled training data $D$; Training epoch number $\tau$; Word vector distance matrix $W$;
**Output**: Trained target CNN model $\theta$;
**Initialisation**: t=1; Randomly initialise $\theta$;
**while** $t \le \tau$ **do**
    Computer predictions of model (Eq. 1);
    Normalize the distance to soft targets $\{s_j\}_{j=0}^{C}$ in distance matrix $W$ (Eq.7);
    Distil knowledge from soft targets to the model (Eq. 8);
    Compute the final FFDWV loss function (Eq. 9);
    Update the model parameters $\theta$ by a SGD algorithm.
**end**
```

Figure 3: Example images from (a) CIFAR10 and (b) CIFAR100.

4. Experiments

4.1. Datasets

We used two multi-class categorisation benchmark for experimental evaluations (Figure 3). (1) CIFAR10[24]: This natural dataset contains each 6000 images for 10 object classes (60000 images in total). There are 50000 and 10000 images in the training and test sets respectively, at the size of 32×32. (2) CIFAR100[24]: Similar to CIFAR10, CIFAR100 contains 60000 images of 32x32 pixels in size and has the same division of training/ test set. It has 100 categories, each of class corresponds to 600 images.

4.2. Performance Metrics

We choose the top 1 error rate which is common in image classification task. The loss of training and testing is evaluated by floating point operation (FLOPs) standard.

4.3. Experiment Setup

We build the network and train the model in Pytorch. For the model training, in order to ensure a fair comparison of experiments, we use the same setting as[25-26]. For the training epochs, we set to 300 in both CIFAR10 and CIFAR100. We used the standard learning rate decay scheme, which dropped from 0.1 to 0.01 at 50% point of the whole training process, and to 0.001 at 75% point until the end. And use the SGD with Nesterov momentum and set the momentum to 0.9.

4.4. Experiments Results

4.4.1 Results On Cifar10 And Cifar100

The results in Table 2 is the top-1 error rate comparison of our FFDWV method in two representative machine learning network models. From TABLE II, we obtain the following observations: 1) The proposed FFDWV approach clearly surpasses the vanilla method which without the soft label constrain. 2) We observed similar error rate degradation on both ResNet-32 and ResNet-110[11]. This indicates the advantages and superiority of our method in training variant of deep classification models.

4.4.2 Comparison with Distillation Methods

We compared FFDWV method with two representative knowledge distillation methods: Knowledge Distillation (KD)[13] and Deep Mutual Learning (DML)[27]. For KD, we choose ResNet-110 as teacher model and a small network ResNet-32 as the student. In DML, the network of student and teacher are identical, e.g. either ResNet-32 or ResNet-110. TABLE III shows that our FFDWV is competitive compare to the alternative KD methods in term of classification accuracy. In general, KD and DML slightly outperform our FFDWV, this is expected since the KD and DML deploys a more powerful teacher model to induce the inter-class correlation knowledge. While the knowledge can be more reliable and complete, this is at a price of significantly high computational costs for model optimisation. Therefore, the KD and DML is highly inferior in terms of computational scalability. Overall, these evidences above indicate a superior trade-off between model accuracy and training costs by the proposed method on the image classification problems.

Table 2: Evaluation of our FFDWV Method on CIFAR10 and CIFAR100.

Method	CIFAR10	CIFAR100	Params
ResNet-32[11]	6.93	31.18	0.5M
ResNet-32+**FFDWV**	**6.6±0.07**	**30.3±0.05**	0.5M
ResNet-110[11]	5.56	25.33	1.7M
ResNet-110+**FFDWV**	**5.12±0.04**	**24.75±0.05**	1.7M

Metric: Error rate (%)

Table 3: Comparison with Knowledge Distillation Methods on CIFAR100.

Target Network	ResNet-32			ResNet-110		
Metric	Error(%)	TrCost	TeCost	Error(%)	TrCost	TeCost
KD	28.83[13]	6.43	1.38	N/A	N/A	N/A
DML	29.03±0.22[27]	2.76	1.38	24.10±0.72	10.10	5.05
FFDWV	30.3±0.05	**0.97**	1.38	24.75±0.05	**0.59**	5.05

"*": Reported results. TrCost/TeCost: Training/test cost, in unit of 10^8FLOPs

5. Conclusions

In the work, we propose a free-teacher framework driven by word-vector for knowledge distillation. Compared to existing KD methods, our approach does not require a heavy pre-trained teacher with expensive cost, which suit for the computational resource limited scenarios. Extensive experiments show our method is competitive compared to conventional KD methods.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, 2015, pp. 436-444.

[2] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, A. Krizhevsky, I. Sutskever, and G. E. Hinton, Eds. New York, NY: Association for Computing Machinery, 2012, pp. 1097-1105.

[3] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," Advances in neural information processing systems, Neural Information Processing Systems Foundation, pp. 2017-2025.

[4] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Jun. 2015, pp. 1-9, doi:10.1109/CVPR.2015.7298594.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," European conference on computer vision, Springer, pp. 630-645.

[6] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, Dec. 2015, pp. 1440-1448, doi:10.1109/ICCV.2015.169.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Jun. 2015, pp. 3431-3440, doi:10.1109/CVPR.2015.7298965.

[8] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," Convolutional Neural Networks for Visual Recognition,

[9] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, 2017, pp. 652-663, doi: 10.1109/TPAMI.2016.2587640.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE conference on computer vision and pattern recognition, IEEE, Jun. 2016, pp. 770-778, doi:10.1109/CVPR.2016.90.

[12] S. Zagoruyko and N. Komodakis, "Wide residual networks," arXiv preprint arXiv:1605.07146, 2016, doi: 10.5244/C.30.87.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[14] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," arXiv preprint arXiv:1412.6550, 2014.

[15] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," arXiv preprint arXiv:1612.03928, 2016.

[16] Google Code. (2013). word2vec [online]. Available: https://code.google.com/archive/p/word2vec/

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[18] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Advances in Neural Information Processing Systems pp. 3111-3119.

[19] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 746-751.

[20] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 535-541,

[21] J. Ba and R. Caruana, "Do deep nets really need to be deep?," Advances in neural information processing systems pp. 2654-2662.

[22] W.M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, "Sobolev training for neural networks," Advances in Neural Information Processing Systems pp. 4278-4287.

[23] P.-T. De Boer, D.P. Kroese, S. Mannor, and R.Y. Rubinstein, "A tutorial on the cross-entropy method," Annals of Operations Research, vol. 134, no. 1, 2005, pp. 19-67.

[24] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Technical Report TR-2009 2009.

[25] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," Proceedings of the IEEE conference on computer vision and pattern recognition pp. 1492-1500.

[26] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K.Q. Weinberger, "Deep networks with stochastic depth," European conference on computer vision, Springer, pp. 646-661.

[27] Y. Zhang, T. Xiang, T.M. Hospedales, and H. Lu, "Deep mutual learning," arXiv e-print, 2017.