Knowledge Transfer via Dense Cross-Layer Mutual-Distillation

Skorik Sergey

MIPT, 2023

May 16, 2023

1 Method

2 Experiments

Knowledge distillation

Formulation

Review the formulation of Knowledge Distillation (KD).
Given the training data $X = \{x_n\}_{n=1}^N$ and , the ground-truth labels are denoted as $Y = \{y_n\}_{n=1}^N$. Let W_t be a teacher network trained beforehand and fixed, and let W_s be a student model. In KD, the student network W_s is trained by minimizing

$$L_s = L_c(W_s, X, Y) + \lambda L_{kd}(\hat{P}_t, \hat{P}_s) \tag{1}$$

Where L_c classification loss by hard labels, L_{kd} is the distillation loss

$$L_{kd}(\hat{P}_t, \hat{P}_s) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{M}\hat{P}_t^m(x_n)\log\hat{P}_s^m(x_n) \tag{2}$$

Deep Mutual Learning

Formulation

DML can be viewed as a bidirectional KD method that jointly trains the teacher and student networks via interleavingly optimizing two objectives:

$$\begin{aligned} L_s &= L_c(W_s, X, Y) + \lambda L_{dml}(\hat{P}_t, \hat{P}_s) \\ L_t &= L_c(W_t, X, Y) + \lambda L_{dml}(\hat{P}_s, \hat{P}_t) \end{aligned} \tag{3}$$

instead of using (2), DML uses Kullback-Leibler divergence:

$$L_{dml}(\hat{P}_t, \hat{P}_s) = \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} \hat{P}_t^m(x_n) \log \frac{\hat{P}_t^m(x_n)}{\hat{P}_s^m(x_n)} \tag{4}$$

Dense Cross-Layer Mutual-Distillation

Formulation

Let $Q = \{(t_k, s_k)\}_{k=1}^{K}$ be a set containing K pairs of the same-staged layer indices of the teacher network W_t and the student network W_s, indicating the locations where auxiliary classifiers are added. Let (t_{K+1}, s_{K+1}) indicating the head classifiers. DCM simultaneously minimizes the following two objectives:

$$
\begin{aligned}
L_s &= L_c(W_s, X, Y) + \alpha L_{ds}(W_s, X, Y) + \beta L_{dcm_1}(\hat{P}_t, \hat{P}_s) + \gamma L_{dcm_2}(\hat{P}_t, \hat{P}_s) \\
L_t &= L_c(W_t, X, Y) + \alpha L_{ds}(W_t, X, Y) + \beta L_{dcm_1}(\hat{P}_s, \hat{P}_t) + \gamma L_{dcm_2}(\hat{P}_s, \hat{P}_t)
\end{aligned} \tag{5}
$$

Dense Cross-Layer Mutual-Distillation

Formulation

L_{ds} denotes the total cross-entropy loss over all auxiliary classifiers added to the different-staged layers of the student network, which is computed as

$$L_{ds}(W_s, X, Y) = \sum_{k=1}^{K} L_c(W_{s_k}, X, Y) \tag{6}$$

L_{dcm_1}, L_{dcm_2} denotes the total loss of the same-staged and different-staged bidirectional KD operations respectevily, which is defined as

$$L_{dcm_1}(\hat{P}_t, \hat{P}_s) = \sum_{k=1}^{K} L_{kd}(\hat{P}_{t_k}, \hat{P}_{s_k}) \tag{7}$$

$$L_{dcm_2}(\hat{P}_t, \hat{P}_s) = \sum_{\{(i,j):1 \leqslant i,j \leqslant K+1, i \neq j\}} L_{kd}(\hat{P}_{t_i}, \hat{P}_{s_j}) \tag{8}$$

Dense Cross-Layer Mutual-Distillation

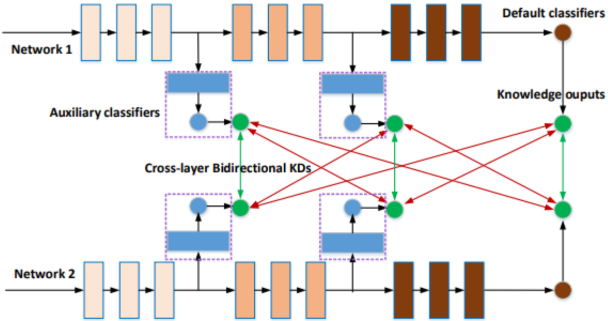

Figure: Structure overview of the proposed method.

Dense Cross-Layer Mutual-Distillation

Algorithm 1: The DCM algorithm

Input : Training data $\{X, Y\}$, two CNN models W_t and W_s, classifier locations $\{(t_k, s_k)\}_{k=1}^{K+1}$, learning rate γ_i

Initialise W_t and W_s, $i = 0$;

repeat

$i \leftarrow i + 1$, update γ_i;

1. Randomly sample a batch of data from $\{X, Y\}$;

2. Compute knowledge set $\{(\hat{P}_{t_k}, \hat{P}_{s_k})\}_{k=1}^{K+1}$ at all supervised layers of two models by Eq. 3;

3. Compute loss L_t and L_s by Eq. 6, Eq. 7, Eq. 8, and Eq. 9 ;

4. Calculate gradients and update parameters:

$W_t \leftarrow W_t - \gamma_i \frac{\partial L_t}{\partial W_t}$, $W_s \leftarrow W_s - \gamma_i \frac{\partial L_s}{\partial W_s}$

until *Converge*;

Experiments on CIFAR-100

Table 1. Result comparison on the CIFAR-100 dataset. WRN-28-10(+) denotes the models trained with dropout. Bolded results show the accuracy margins of DCM compared to DML. *In this paper, for each joint training case on the CIFAR-100 dataset, we run each method 5 times and report "mean(std)" top-1 error rates (%). Results of all methods are obtained with the exactly same training hyper-parameters, and our CNN baselines mostly have better accuracies compared to the numbers reported in their original papers [13,19,52,57].*

Networks		Ind(baseline)		DML		DCM	
Net1	Net2	Net1	Net2	Net1	Net2	Net1|**DCM-DML**	Net2|**DCM-DML**
ResNet-164	ResNet-164	22.56(0.20)	22.56(0.20)	20.69(0.25)	20.72(0.14)	19.57(0.20)|**1.12**	19.59(0.15)|**1.13**
WRN-28-10	WRN-28-10	18.72(0.24)	18.72(0.24)	17.89(0.26)	17.95(0.07)	16.61(0.24)|**1.28**	16.65(0.22)|**1.30**
DenseNet-40-12	DenseNet-40-12	24.91(0.18)	24.91(0.18)	23.18(0.18)	23.15(0.20)	22.35(0.12)|**0.83**	22.41(0.17)|**0.74**
WRN-28-10	ResNet-110	18.72(0.24)	26.55(0.26)	17.99(0.24)	24.42(0.19)	17.82(0.14)|**0.17**	22.99(0.30)|**1.43**
WRN-28-10	WRN-28-4	18.72(0.24)	21.39(0.30)	17.80(0.11)	20.21(0.16)	16.84(0.08)|**0.96**	18.76(0.14)|**1.45**
WRN-28-10	MobileNet	18.72(0.24)	26.30(0.35)	17.24(0.13)	23.91(0.22)	16.83(0.07)|**0.41**	21.43(0.20)|**2.48**
WRN-28-10(+)	WRN-28-10(+)	18.64(0.19)	18.64(0.19)	17.62(0.12)	17.61(0.13)	16.57(0.12)|**1.05**	16.59(0.15)|**1.02**

Experiments on ImageNet

Table 2. Result comparison on the ImageNet classification dataset. For each network, we report top-1/top-5 error rate (%). Bolded results show the accuracy margins of DCM compared to the independent training method/DML.

Networks		Ind(baseline)		DML		DCM					
Net1	Net2	Net1	Net2	Net1	Net2	Net1	**DCM-Ind**	**DCM-DML**	Net2	**DCM-Ind**	**DCM-DML**
ResNet-18	ResNet-18	31.08/11.17	31.08/11.17	29.13/9.89	29.25/10.00	28.67/9.71	**2.41/1.46**	**0.46/0.18**	28.74/9.74	**2.34/1.43**	**0.51/0.26**
MobileNetV2	MobileNetV2	27.80/9.50	27.80/9.50	26.61/8.85	26.78/8.97	25.62/8.16	**2.18/1.34**	**0.99/0.69**	25.74/8.21	**2.06/1.29**	**1.04/0.76**
ResNet-50	ResNet-18	25.47/7.58	31.08/11.17	25.24/7.56	28.65/9.49	24.92/7.42	**0.55/0.16**	**0.32/0.14**	27.93/9.19	**3.15/1.98**	**0.72/0.30**

Deep Analysis of DCM

Summary

- Variation of the location and structure of auxiliary classifiers can give an improvement (specifying the Q set)
- L_{dcm_2} loss term gives more impact on accuracy.
- A simple addition of classifiers to the DML without using the objective terms L_{dcm_1}, L_{dcm_2} or to the baseline method with their independent training does not bring a significant improvement. Thus, the improvement of DCM in comparison with other methods is directly related to the learning mechanism, and not due to an increase in the number of neural network parameters.