Deep Mutual Learning

Ying Zhang[1,2], Tao Xiang[2], Timothy M. Hospedales[3], and Huchuan Lu[1]
[1] Dalian University of Technology, China
[2] Queen Mary University of London, UK [3] University of Edinburgh, UK
zydl0907@mail.dlut.edu.cn, t.xiang@qmul.ac.uk, t.hospedales@ed.ac.uk, lhchuan@dlut.edu.cn

Abstract

Model distillation is an effective and widely used technique to transfer knowledge from a teacher to a student network. The typical application is to transfer from a powerful large network or ensemble to a small network, in order to meet the low-memory or fast execution requirements. In this paper, we present a deep mutual learning (DML) strategy. Different from the one-way transfer between a static pre-defined teacher and a student in model distillation, with DML, an ensemble of students learn collaboratively and teach each other throughout the training process. Our experiments show that a variety of network architectures benefit from mutual learning and achieve compelling results on both category and instance recognition tasks. Surprisingly, it is revealed that no prior powerful teacher network is necessary – mutual learning of a collection of simple student networks works, and moreover outperforms distillation from a more powerful yet static teacher.

1. Introduction

Deep neural networks achieve state of the art performance on many problems, but are often very large in depth and/or width, and contain large numbers of parameters [7, 28]. This has the drawback that they may be slow to execute or large to store, limiting their use in applications or platforms with low memory or fast execution requirements, e.g., mobile phones. This has led to a rapid growth of research in developing smaller and faster models. Achieving compact yet accurate models has been approached in a variety of ways including explicit frugal architecture design [9], model compression [22], pruning [14], binarisation [18] and most interestingly model distillation [8].

Distillation-based model compression relates to the observation [3, 1] that small networks often have the same *representation capacity* as large networks; but compared to large networks they are simply harder to train and find the right parameters that realise the desired function. That is, the limitation seems to lie in the difficulty of optimisation rather than in the network size [1]. To better learn a small network, the distillation approach starts with a powerful (deep and/or wide) teacher network (or network ensemble), and then trains a smaller student network to *mimic* the teacher [8, 1, 16, 3]. Mimicking the teacher's class probabilities [8] and/or feature representation [1, 19] conveys additional information beyond the conventional supervised learning target. The optimisation problem of learning to mimic the teacher turns out to be easier than learning the target function directly, and the student can match or even outperform [19] the much larger teacher.

In this paper we aim to solve the same problem of learning small but powerful deep neural networks. However, we explore a different but related idea to model distillation – that of *mutual learning*. Distillation starts with a powerful large and pre-trained teacher network and performs one-way knowledge transfer to a small untrained student. In contrast, in mutual learning we start with a pool of untrained students who simultaneously learn to solve the task together. Specifically, each student is trained with two losses: a conventional supervised learning loss, and a mimicry loss that aligns each student's class posterior with the class probabilities of other students. Trained in this way, it turns out that each student in such a peer-teaching based scenario learns significantly better than when learning alone in a conventional supervised learning scenario. Moreover mutually learned student networks achieve better results than students trained by conventional distillation from a larger pre-trained teacher. Furthermore, while the conventional understanding of distillation requires a teacher larger and more powerful than the intended student, it turns out that in many cases mutual learning of several large networks also improves performance compared to independent learning. This makes the deep mutual learning strategy generally applicable, e.g., it can also be used in application scenarios where there is no constraint on the model size and the recognition accuracy is the only concern.

It is perhaps not obvious why the proposed learning strategy should work at all. Where does the additional knowledge come from, when the learning process starts out with

all small and untrained student networks? Why does it converge to a good solution rather than being hamstrung by groupthink as 'the blind lead the blind'. Some intuition about these questions can be gained by considering the following: Each student is primarily directed by a conventional supervised learning loss, which means that their performance generally increases and they cannot drift arbitrarily into groupthink as a cohort. With supervised learning, all networks soon predict the same (true) labels for each training instance; but since each network starts from a different initial condition, they learn different representations, and consequently their estimates of the probabilities of the next most likely classes vary. It is these secondary quantities that provide the extra information in distillation [8] as well as mutual learning. In mutual learning the student cohort effectively pools their collective estimate of the next most likely classes. Finding out – and matching – the other most likely classes for each training instance according to their peers increases each student's posterior entropy [4, 17], which helps them to converge to a more robust (flatter) minima with better generalisation to testing data. This is related to very recent work on the robustness of high posterior entropy solutions (network parameter settings) in deep learning [4, 17], but with a more informed choice of alternatives than blind entropy regularisation.

Overall, mutual learning provides a simple but effective way to improve the generalisation ability of a network by training collaboratively with a cohort of other networks. Extensive experiments are carried out on both object category recognition (image classification on CIFAR100 [12]) and instance recognition problems (person re-identiciation on Market1501 [33]). The results show that, compared with distillation by a pre-trained static large network, collaborative learning by small peers achieves better performance. In particular, on the person re-identification task, state-of-the-art results can be obtained using a much smaller network trained with mutual learning, compared to the latest competitors. Furthermore we observe that: (i) it applies to a variety of network architectures, and to heterogeneous cohorts consisting of mixed big and small networks; (ii) The efficacy increases with the number of networks in the cohort – a nice property to have because by training on small networks only, more of them can fit on given GPU resources for more effective mutual learning; (iii) it also benefits semi-supervised learning with the mimicry loss activated both on labelled and unlabelled data. Finally, we note that while our focus is on obtaining a single effective network, the entire cohort can also be used as a highly effective ensemble model.

2. Related Work

Model Distillation The distillation-based approach to model compression has been proposed over a decade ago [3] but was recently re-popularised by [8], where some additional intuition about why it works – due to the additional supervision and regularisation of the higher entropy soft-targets – was presented. Initially, a common application was to distill the function approximated by a powerful model/ensemble teacher into a single neural network student [3, 8]. But later, the idea has been applied to distill powerful and easy-to-train large networks into small but harder-to-train networks [19] that can even outperform their teacher. Recently, distillation has been connected more systematically to information learning theory [15] and SVM+ [25] – an intelligent teacher provides privileged information to the student. This idea of using model distillation for learning with privileged information has been exploited by Zhang *et al*. [29] for action recognision: the more expensive optical flow field is treated as privileged information and an optical flow CNN is used to teach a motion vector CNN. In terms of representation of the knowledge to be distilled from the teacher, existing models typically use teacher's class probabilities [8] and/or feature representation [1, 19]. Recently, Yim *et al*. [27] exploited flow between layers computed as the inner product of feature maps between layers. In contrast to model distillation, we address dispensing with the teacher altogether, and allowing an ensemble of students to teach each other in mutual distillation.

Collaborative Learning Other related ideas on collaborative learning include Dual Learning [6] where two cross-lingual translation models teach each other interactively. But this only applies in this special translation problem where an unconditional within-language model is available to be used to evaluate the quality of the predictions, and ultimately provides the supervision that drives the learning process. Furthermore, in dual learning different models have different learning tasks whilst in mutual learning the tasks are identical. Recently, Cooperative Learning [2] has been proposed to learn multiple models jointly for the same task but in different domains. E.g. recognising the same set of object categories but with one model inputting RGB images and the other inputting depth images. The models communicate via object attributes which are domain invariant. Again this is different from mutual learning where all models address the same task and domain.

3. Deep Mutual Learning

3.1. Formulation

We formulate the proposed deep mutual learning (DML) approach with a cohort of two networks (see Fig. 1). Extension to more networks is straightforward (see Sec. 3.3). Given N samples $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ from M classes, we denote the corresponding label set as $\mathcal{Y} = \{y_i\}_{i=1}^N$ with $y_i \in \{1, 2, ..., M\}$. The probability of class m for sample

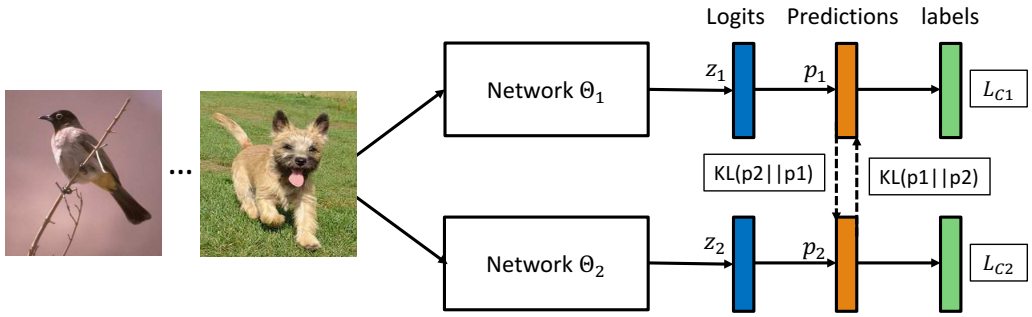


Figure 1. Deep Mutual Learning (DML) schematic. Each network is trained with a supervised learning loss, and a Kullback Leibler Divergence based mimcry loss to match the probability estimates of its peers.

$\boldsymbol{x}_i$ given by a neural network Θ_1 is computed as

$$p_1^m(\boldsymbol{x}_i) = \frac{exp(z_1^m)}{\sum_{m=1}^{M} exp(z_1^m)}, \tag{1}$$

where the logit z^m is the output of the "softmax" layer in Θ_1.

For multi-class classification, the objective function to train the network Θ_1 is defined as the cross entropy error between the predicted values and the correct labels,

$$L_{C_1} = -\sum_{i=1}^{N}\sum_{m=1}^{M} I(y_i, m)\log(p_1^m(\boldsymbol{x}_i)), \tag{2}$$

with an indicator function I defined as

$$I(y_i, m) = \begin{cases} 1 & y_i = m \\ 0 & y_i \neq m \end{cases} \tag{3}$$

The conventional supervised loss trains the network to predict the correct labels for the training instances. To improve the generalisation performance of Θ_1 on the testing instances, we use another peer network Θ_2 to provide training experience in the form of its posterior probability $\boldsymbol{p}_2$. To quantify the match of the two network's predictions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, we use the Kullback Leibler (KL) Divergence.

The KL distance from $\boldsymbol{p}_1$ to $\boldsymbol{p}_2$ is computed as

$$D_{KL}(\boldsymbol{p}_2\|\boldsymbol{p}_1) = \sum_{i=1}^{N}\sum_{m=1}^{M} p_2^m(\boldsymbol{x}_i)\log\frac{p_2^m(\boldsymbol{x}_i)}{p_1^m(\boldsymbol{x}_i)}. \tag{4}$$

The overall loss functions L_{Θ_1} and L_{Θ_2} for networks Θ_1 and Θ_2 respectively are thus:

$$L_{\Theta_1} = L_{C_1} + D_{KL}(\boldsymbol{p}_2\|\boldsymbol{p}_1). \tag{5}$$
$$L_{\Theta_2} = L_{C_2} + D_{KL}(\boldsymbol{p}_1\|\boldsymbol{p}_2). \tag{6}$$

In this way each network learns both to correctly predict the true label of training instances (supervised loss L_C) as well as to match the probability estimate of its peer (KL mimicry loss).

Our KL divergence based mimicry loss is asymmetric, thus different for the two networks. One can instead use a symmetric Jensen-Shannon Divergence loss:

$$\frac{1}{2}(D_{KL}(\boldsymbol{p}_1\|\boldsymbol{p}_2) + D_{KL}(\boldsymbol{p}_1\|\boldsymbol{p}_2)). \tag{7}$$

However, we found empirically that whether a symmetric or asymmetric KL loss is used does not make any difference.

3.2. Optimisation

A key difference between model distillation and DML is that in DML, the two models are optimised jointly and collaboratively, with the optimisation processes for the two models being closely intervened. The mutual learning strategy is embedded in each mini-batch based model update step for both models and throughout the whole training process. The models are learned with the same mini-batches. At each iteration, we compute the predictions of the two models and update both networks' parameters according to the predictions of the other. The optimisation of Θ_1 and Θ_2 is conducted iteratively until convergence. The optimisation details are summarised in Algorithm 1. It consists of 4 sequential steps if running on a single GPU. When two GPUs are available, distributed training can be implemented by running Steps 1, 2 on one GPU and Steps 3,4 on another in parallel.

3.3. Extension to Larger Student Cohorts

The proposed DML approach naturally extends to more networks in the student cohort. Given K networks $\Theta_1, \Theta_2, ..., \Theta_K (K \geq 2)$, the objective function for optimising $\Theta_k, (1 \leq k \leq K)$ becomes

$$L_{\Theta_k} = L_{C_k} + \frac{1}{K-1}\sum_{l=1, l\neq k}^{K} D_{KL}(\boldsymbol{p}_l\|\boldsymbol{p}_k). \tag{10}$$

Equation (10) indicates that with K networks, DML for each student effectively takes the other $K-1$ networks in the cohort as teachers to provide mimicry targets. Equation (6) is now a special case of (10) with $K = 2$. Note that

Algorithm 1: Deep Mutual Learning

Input: Training set $\mathcal{X}$, label set $\mathcal{Y}$, learning rate γ_t
Initialize: Initialise Θ_1 and Θ_2 to different conditions; $t = 0$.
Repeat :
$t = t + 1$
Randomly sample data $\boldsymbol{x}$ from $\mathcal{X}$.
Compute predictions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ by (1).
1: Compute the stochastic gradient and update Θ_1 :

$$\Theta_1 \leftarrow \Theta_1 + \gamma_t \frac{\partial L_{\Theta_1}}{\partial \Theta_1} \tag{8}$$

2: Update the predictions $\boldsymbol{p}_1$ of $\boldsymbol{x}$ by (1).
3: Compute the stochastic gradient and update Θ_2 :

$$\Theta_2 \leftarrow \Theta_2 + \gamma_t \frac{\partial L_{\Theta_2}}{\partial \Theta_2} \tag{9}$$

4: Update the predictions $\boldsymbol{p}_2$ of $\boldsymbol{x}$ by (1).
Until : convergence

we have added the coefficient $\frac{1}{K-1}$ to make sure that the training is mainly directed by supervised learning of the true labels. The optimisation for DML with more than two networks is a straightforward extension of Algorithm 1. This learning strategy naturally suits distributed learning [21]: It can be distributed by learning each network on one device and passing the small probability vectors between devices.

With more than two networks, an interesting alternative learning strategy for DML is to take the ensemble of all the other $K-1$ networks as a single teacher to provide a combined mimicry target, which would be very similar to the distillation approach but performed at each mini-batch model update. Then the objective function of Θ_k can be written as

$$L_{\Theta_k} = L_{C_k} + D_{KL}(\boldsymbol{p}_{avg} \| \boldsymbol{p}_k), \quad \boldsymbol{p}_{avg} = \frac{1}{K-1} \sum_{l=1, l \neq k}^{K} \boldsymbol{p}_l. \tag{11}$$

In our experiments (see Sec. 4.8), we find that this DML strategy with a single ensemble teacher (denoted DML_e) leads to worse performance than DML with $K-1$ teachers. This is because the model averaging step (Equation (11)) to build the teacher ensemble makes the teacher's posterior probabilities more peaked at the true class, thus reducing the posterior entropy over all classes. It is therefore contradictory to one of the objectives of DML which is to produce robust solutions with high posterior entropy.

3.4. Extension to Semi-supervised Learning

The proposed DML extends straightforwardly to semi-supervised learning. Under the semi-supervised learning setting, we only activate the cross-entropy loss for labelled data, while computing the KL distance based mimicry loss for all the training data. This is because the KL distance computation does not require class labels, so unlabelled data can also be used. Denote the labelled and unlabelled data as $\mathcal{L}$ and $\mathcal{U}$, where we have $\mathcal{X} = \mathcal{L} \cup \mathcal{U}$, the objective function for learning network Θ_1 can be reformulated as

$$L_{\Theta_1} = \underset{\boldsymbol{x} \in \mathcal{L}}{L_{C_1}} + \underset{\boldsymbol{x} \in \mathcal{X}}{D_{KL}}(\boldsymbol{p}_2 \| \boldsymbol{p}_1). \tag{12}$$

4. Experiments

4.1. Datasets and Settings

Datasets Four datasets are used in our experiments. The **ImageNet** [20] dataset contains 1000 object classes with about 1.2 million images for training and 50,000 images for validation. The **CIFAR-10** and **CIFAR-100** [12] datasets consist of 32×32 color images containing objects from 10 and 100 classes respectively. Both are split into a 50,000-image train set and a 10,000-image test set. The Top-1 classification accuracy is used as evaluation metric. The **Market-1501** [33] dataset is a widely used benchmark in the person re-identification (re-id) problem [5]. Different from the object category recognition problem in CIFAR, re-id is an instance recognition problem that aims to associate person identities across different non-overlapping camera views. Market-1501 contains 32,668 images of 1,501 identities captured from six camera views, with 751 identities for training and 750 identities for testing. As per state of the art approaches to re-id [35], we train the network for 751-way classification and use the resulting feature output of the last pooling layer as a representation for nearest neighbour matching at testing. For evaluation, the standard Cumulative Matching Characteristic (CMC) Rank-k accuracy and mean average precision (mAP) metrics [33] are used.

Networks The networks used in our experiments include compact networks of typical student size: Resnet-32 [7] and MobileNet [9]; as well as large networks of typical teacher size: InceptionV1 [24] and Wide ResNet WRN-28-10 [28]. Table 1 compares the number of parameters of all the networks for CIFAR-100.

ResNet-32	MobileNet	InceptionV1	WRN-28-10
0.5M	3.3M	7.8M	36.5M

Table 1. Number of parameters on the CIFAR-100 dataset

Implementation Details We implement all networks and training procedures in TensorFlow and conduct all experiments on an NVIDIA GeForce GTX 1080 GPU. For CIFAR-100, we follow the experimental settings of [28]. Specifically, we use SGD with Nesterov momentum and set the initial learning rate to 0.1, momentum to 0.9 and mini-batch size to 64. The learning rate dropped by 0.1 every 60 epochs and we train for 200 epochs. The data augmentation includes horizontal flips and random crops from im-

Network Types		CIFAR-10						CIFAR-100					
		Independent		DML		DML-Ind		Independent		DML		DML-Ind	
Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2
Resnet-32	Resnet-32	92.47	92.47	92.68	92.80	0.21	0.33	68.99	68.99	71.19	70.75	2.20	1.76
WRN-28-10	Resnet-32	95.01	92.47	95.75	93.18	0.74	0.71	78.69	68.99	78.96	70.73	0.27	1.74
MobileNet	Resnet-32	93.59	92.47	94.24	93.32	0.65	0.85	73.65	68.99	76.13	71.10	2.48	2.11
MobileNet	MobileNet	93.59	93.59	94.10	94.30	0.51	0.71	73.65	73.65	76.21	76.10	2.56	2.45
WRN-28-10	MobileNet	95.01	93.59	95.73	94.37	0.72	0.78	78.69	73.65	80.28	77.39	1.59	3.74
WRN-28-10	WRN-28-10	95.01	95.01	95.66	95.63	0.65	0.62	78.69	78.69	80.28	80.08	1.59	1.39

Table 2. Top-1 accuracy (%) on the CIFAR-10 and CIFAR-100 dataset. "DML-Ind" measures the difference in accuracy between the network learned with DML and the same network learned independently.

age padded by 4 pixels on each side, filling missing pixels with reflections of original image. For Market-1501, we use the Adam optimiser [11], with learning rate $lr = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a mini-batch size of 16. For ImageNet, we use RMSProp with decay of 0.9, mini-batch size of 64, and initial learning rate of 0.1. The learning rate decayed every 20 epochs using an exponential rate of 0.16.

4.2. Results on CIFAR-100

Table 2 compares the Top-1 accuracy of the CIFAR-100 dataset obtained by various architectures in a two-network DML cohort. From the table we can make the following observations: (i) All the network combinations among ResNet-32, MobileNet and WRN-28-10 improve performance when learning in a cohort compared to learning independently, indicated by the all positive values in the "DML-Independent" columns. (ii) The networks with smaller capacity (ResNet-32 and MobileNet) generally benefit more from DML. (iii) Although WRN-28-10 is a much larger network than MobileNet or ResNet-32 (Table 1), it still benefits from being trained together with a smaller peer. (iv) Training a cohort of large networks (WRN-28-10) is still beneficial compared to learning them independently. Thus in contrast to the conventional wisdom of model distillation, we see that a large pre-trained teacher does not necessary bring large benefits, and multiple large networks can still benefit from our distillation-like process.

4.3. Results on Market-1501

In this experiment, we use MobileNet in a two-network DML cohort. Table 3 summarises the mAP (%) and rank-1 accuracy (%) of Market-1501 of MobileNets trained with/without DML, as well as the comparison against existing state of the art methods. We can see that on this more challenging instance recognition problem, DML greatly improves the performance of MobileNet compared to independent learning, both with and without pre-training on ImageNet. It can also be seen that our DML approach using two MobileNets significantly outperforms prior state-of-the-art deep re-id methods. This is noteworthy as MobileNet is a simple, small, and general-purpose network. In contrast many recently proposed deep re-id networks such as those in [31, 23, 34] have complicated and specially designed architectures to handle the drastic pose-changes and body-part mis-alignment when matching people across camera views.

Method	pre?	Single-Query		Multi-Query	
		mAP	Rank-1	mAP	Rank-1
Gated S-CNN [26]	no	39.55	65.88	48.45	76.04
k-reciprocal [35]	yes	63.63	77.11	-	-
MSCAN [13]	no	57.53	80.31	66.70	86.79
PDC [23]	no	63.41	84.14	-	-
DLPAR [32]	no	63.40	81.00	-	-
MobileNet	no	50.15	76.87	60.16	84.06
MobileNet+DML	no	54.71	79.12	64.10	85.63
MobileNet	yes	65.06	85.01	74.53	90.59
MobileNet+DML	yes	**70.51**	**89.34**	**78.95**	**92.81**

Table 3. Comparative results on the Market-1501 dataset. Each MobileNet is trained in a two-network cohort and the averaged performance of the two networks in the cohort is reported. 'pre?' indicates whether ImageNet pretraining was carried out.

4.4. Results on ImageNet

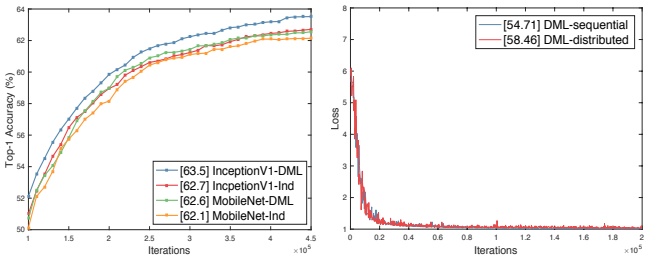
Figure 2 (a) compares MobileNet and InceptionV1 accuracy on ImageNet with Independent and DML training. We can see that the DML variants of both architectures consistently performs better than their independently trained counterparts. These results show that DML is applicable to large-scale problems.

4.5. Distributed Training of DML

To investigate the impact of training strategy on DML, we compared two DML variants: 1) sequential: train two networks according to Algorithm 1 on one GPU; two networks are updated one after the other; 2) distributed: each network is trained on a separate GPU and CPU is used for KL divergence communication; in this way, the predictions and parameters of two networks are updated simultaneously. We experiment on Market-1501 with 2 MobileNets,

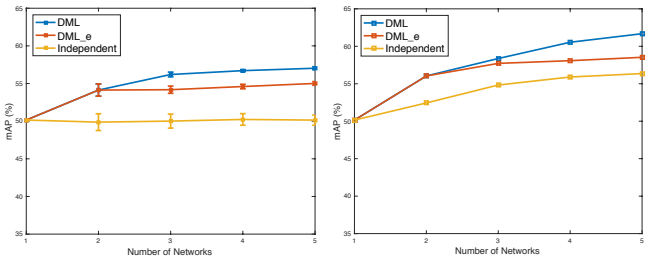
Dataset	Network Types		Independent		1 distills 2	DML	
	Net1	Net 2	Net 1	Net 2	Net 2	Net 1	Net 2
CIFAR-100	WRN-28-10	ResNet-32	78.69	68.99	69.48	78.96	70.73
	MobilNet	ResNet-32	73.65	68.99	69.12	76.13	71.10
Market-1501	Inception V1	MobileNet	63.91	50.15	52.30	64.42	58.47
	MobileNet	MobileNet	50.15	49.87	50.07	55.28	54.13

Table 4. Comparison with distillation on CIFAR-100 (Top-1 accuracy (%)) and Market-1501 dataset (mAP (%))



(a) ImageNet (b) Market-1501

Figure 2. (a) Results on ImageNet. It shows top-1 acc. (%) w.r.t. training steps; (b) Convergence effect and test mAP (%) on Market-1501 with sequential and distributed training.



(a) average mAP (b) ensemble mAP

Figure 3. Performance (mAP (%)) on Market-1501 with different cohort size.

and show the convergence and mAP results in Fig. 2 (b). It is interesting to observe that our DML's performance is further boosted by distributed training. Comparing these two variants, the two networks are more 'equal' when trained distributed as they always have exactly the same number of training iterations. This result thus suggests that the students benefit the most from the DML peer-teaching when the discrepancy in their learning progress is minimised.

4.6. Comparison with Model Distillation

As our method is closely related to model distillation, we next provide a focused comparison to Distillation [8]. Table 4 compares our DML with model distillation where the teacher network (Net 1) is pre-trained and provides fixed posterior targets for the student network (Net 2). As expected the conventional distillation approach from a powerful pre-trained teacher does indeed improve the student performance compared to independently learning the student (1 distills 2 versus Net 2 Independent).

However, the results show that training both networks together in deep mutual learning provides a clear improvement compared to distillation (1 distills 2 versus DML Net 2). This implies that in the process of mutual learning, the network that would play the role of teacher actually becomes better than a pre-trained teacher, via learning from interactions with an a-priori untrained student.

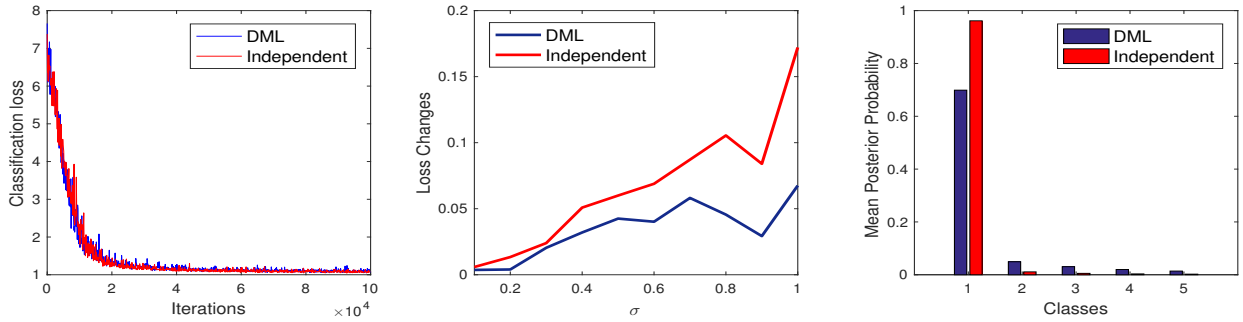
4.7. DML with Larger Student Cohorts

The prior experiments studied cohorts of 2 students. We next investigate how DML scales with more students in the cohort. Figure 3(a) shows the results on Market-1501 with DML training of increasing cohort sizes of MobileNets. The figure shows average mAP, as well as the standard deviation. From Fig. 3(a) we can see that the performance of the average *single* network increases with the number of networks in the DML cohort, hence its gap to the independently trained networks. This demonstrates that the generalisation ability of students is enhanced when learning together with increasing numbers of peers. The performance of different networks is also more consistent with larger cohort size, indicated by the smaller standard deviations.

A common technique when training multiple networks is to use them as an ensemble and make a combined prediction. In Fig. 3(b) we use the same models as Fig. 3(a) but make predictions based on the ensemble (matching based on concatenated feature of all members) instead of reporting the average prediction of each individual. From the results we can see that the ensemble prediction outperforms individual network predictions as expected (Fig. 3(b) vs. (a)). Moreover, the ensemble performance also benefits from training multiple networks as a DML cohort (Fig. 3(b) DML ensemble vs. Independent ensemble).

4.8. How and Why does DML Work?

In this section we attempt to give some insights about how and why our deep mutual learning strategy works. There has been a wave of recent research on the subject of "Why Deep Nets Generalise" [4, 30, 10], which have provided some insights such as: While there are often many solutions (deep network parameter settings) that generate zero training error, some of these generalise better than oth-



(a) Training Loss (b) Loss change given parameter noise (c) Posterior certainty comparison

Figure 4. Analysis on why DML works

ers due to being in wide valleys rather than narrow crevices [4, 10] – so that small perturbations do not change the prediction efficacy drastically; and that deep networks are better than might be expected at finding these good solutions [30], but that the tendency towards finding robust minima can be enhanced by biasing deep nets towards solutions with higher posterior entropy [4, 17].

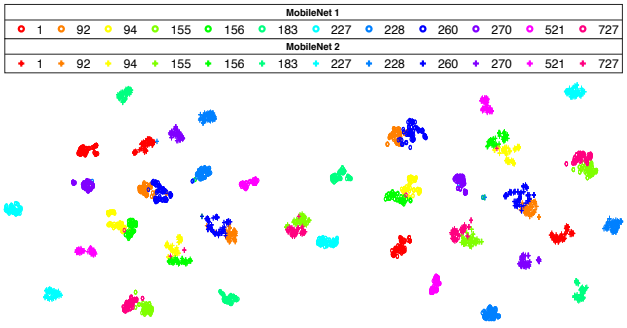
DML Leads to Better Quality Solutions with More Robust Minima With these insights in mind we make some observations about the DML process. Firstly we note that in our experiments, the networks typically fit the training data perfectly: Training accuracy goes to 100% and classification loss becomes minimal (e.g., Fig. 4(a)). However, as we saw earlier, DML performs better on test data. Therefore rather than helping to find a better (deeper) minimum of training loss, DML appears to be helping us to find a wider/more robust minimum that generalises better to test data. Inspired by [4, 10], we perform a simple test to analyse the robustness of the discovered minima on CIFAR-100 using MobileNet. For the DML and independent models, we compare the training loss of the learned models before and after adding independent Gaussian noise with variable standard deviation σ to each model parameter. We see that the depths of the two minima were the same (Fig. 4(a)), but after adding this perturbation the training loss of the independent model jumps up while the loss of the DML model increases much less. This suggests that the DML model has found a much *wider* minimum, which is expected to provide better generalisation performance [4, 17].

How is a Better Minimum Found? When asking each network to match its peer's probability estimates, mismatches where a given network predicts zero and its teacher/peer predicts non-zero are heavily penalised. Therefore the overall effect of DML is that, where each network independently would put a small mass on a small set of secondary probabilities, all networks in the DML tend to aggregate their prediction of secondary probabilities, and both (i) put more mass on the secondary probabilities altogether, and (ii) place non-zero mass on more distinct secondary probabilities. We illustrate this effect by comparing the probabilities assigned to the top-5 highest ranked classes obtained by a ResNet-32 on CIFAR-100 trained by DML vs. an independently trained ResNet-32 model in Fig. 4(c). For each training sample, the top 5 classes are ranked according to the posterior probabilities produced by the model (Class 1 being the true class and Class 2 the second most probable class, etc). Here we can see that the assignment of mass to probabilities below the Top-1 decays more quickly for Independent than DML. This can be quantified by the entropy values averaged over all training samples of the DML trained model and the independently trained model, which are 1.7099 and 0.2602 respectively. Thus our method has connection to entropy regularisation-based approaches [4, 17] to finding wide minima, but by mutual probability matching on 'reasonable' alternatives, rather than a blind high-entropy preference. Table 5 further shows that this is more effective way to learn a more generalisation model when DML is compared with the entropy regularisation-based approach [4] (DML vs. Independent, Entropy).

Settings	mAP	Rank-1
Independent	50.15	76.87
DML	54.71	79.12
Independent, Entropy [4]	50.94	76.34
DML, L2	51.01	77.58

Table 5. Single-Query results on Market-1501 under different settings with MobileNets. L2 indicates decreasing the feature distance of two networks.

DML with Ensemble Teacher In DML, each student is taught by all other students in the cohort individually, regardless how many students are in the cohort (Eq. (10)). In Sec. 3.3, an alternative DML strategy (DML_e) is discussed, by which each student is asked to match the predictions of the ensemble of all other students in the cohort (Eq. (11)). One might reasonably expect this approach to be better: As the ensemble prediction is better than individual predictions, it should provide a cleaner and stronger teach-



(a) Independent (b) DML

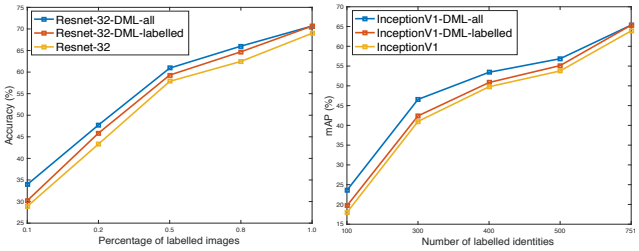
Figure 5. tSNE Visualisation of MobileNets trained with DML and independently on the Market-1501 dataset. Different numbers indicate different identities.

ing signal – more like conventional distillation. In practice the results of ensemble rather than peer teaching are worse (see Fig. 3). By analysing the teaching signal of the ensemble in comparison to peer teaching, the ensemble target is much more sharply peaked at the true label than the peer targets, resulting in larger prediction entropy value for DML (0.2805) than DML_e (0.1562). Thus while the noise-averaging property of ensembling is effective for making a correct prediction, it is actually detrimental to providing a teaching signal where the secondary class probabilities are the salient cue in the signal and having high-entropy posterior leads to more robust solutions to model training.

4.9. Does DML Makes Models More Similar?

We know that with the same training objective, the predicted class posterior would be similar for different models in a DML cohort. The question is do these model also produce similar features, especially when the models have identical architecture? Figure 5 shows the t-SNE visualisation of feature distribution on the Market-1501 test set by two MobileNets. We can see that either with or without DML, the two MobileNets do indeed different features, indicating diverse models are obtained. This helps to explain why different models in a DML cohort can teach each other: Each has learned something that the others have not.

We note that in a number of model distillation studies [1, 19], a feature distance loss is added to force the student network to produce similar features to the teacher at corresponding layers. This makes sense when the teacher is pretrained and fixed and the student aims to imitate the teacher. However, in DML, aligning the internal representations different DML models would diminish the cohort diversity and thus damage the ability of each network to teach its peers. Table 5 shows that indeed, when a feature L2 loss is introduced, DML becomes less effective (DML vs. DML, L2).



(a) CIFAR-100 (b) Market-1501

Figure 6. Semi-supervised learning with DML.

4.10. Semi-Supervised Learning

We finally explore semi-supervised learning in CIFAR-100 and Market-1501. For CIFAR-100 , we randomly select a subset (from 10% to 100%) of the training images per class as labeled, and treat the rest as unlabeled. For Market-1501, we randomly select M identities as labelled in the training set, varying M from 100 to 751. Experiments are performed with 3 different training strategies: 1) training on the labelled data only with single network; 2) training on labelled data only with DML (DML-labelled). 3) training on all data with DML, where the classification loss is computed for labelled data only, and KL loss is calculated for all the training data (DML-all).

From the results in Figure 6, we can see that: (1) Training two networks with DML consistently performs better than training a single network – as before, but now with varying amounts of labeled data. (2) Compared with adding DML to only labelled data (DML-labelled), DML-all further improves the performance by exploiting the unlabelled data using the KL-distance based mimicry loss. The improvement is bigger when the percentage of labelled data is smaller. This confirms that DML benefits both supervised and semi-supervised learning scenarios.

5. Conclusion

We have proposed a simple and generally applicable approach to improving the performance of deep neural networks by training them in a cohort with peers and mutual distillation. With this approach we can obtain compact networks that perform better than those distilled from a strong but static teacher. One application of DML is to obtain compact, fast and effective networks. We also showed that this approach is also promising to improve the performance of large powerful networks, and that the network cohort trained in this manner can be combined as an ensemble to further improve performance.

Acknowledgements. This work was supported by the Natural Science Foundation of China under Grant 61725202 , 61472060 and China Scholarship Council (CSC) and the European Union's Horizon 2020 research and innovation program (grant agreement no. 640891)

References

[1] J. Ba and R. Caruana. Do deep nets really need to be deep? In *NIPS*. 2014.

[2] T. Batra and D. Parikh. Cooperative learning with visual attributes. *arXiv: 1705.05512*, 2017.

[3] C. Bucila, R. Caruana, and A. Niculescu-Mizil. Model compression. In *KDD*, 2006.

[4] P. Chaudhar, A. Choromansk, S. Soatt, Y. LeCun, C. Baldass, C. Borg, J. Chays, L. Sagun, and R. Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR*, 2017.

[5] S. Gong, M. Cristani, S. Yan, and C. C. Loy, editors. *Person Re-Identification*. Advances in Computer Vision and Pattern Recognition. Springer, 2014.

[6] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, and W. Ma. Dual learning for machine translation. In *NIPS*, pages 820–828, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[8] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv: 1503.02531*, 2015.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv: 1704.04861*, 2017.

[10] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[12] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research), 2009.

[13] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning deep context-aware features over body and latent parts for person re-identificatio. In *CVPR*, 2017.

[14] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. In *ICLR*, 2017.

[15] D. Lopez-Paz, L. Bottou, B. Scholkopf, and V. Vapnik. Unifying distillation and privileged information. In *ICLR*, 2016.

[16] E. Parisotto, J. L. Ba, and R. Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. In *ICLR*, 2016.

[17] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR Workshops*, 2017.

[18] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, 2016.

[19] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[21] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Allerton Conference on Communication, Control, and Computing*, pages 909–910, 2015.

[22] W. J. D. Song Han, Huizi Mao. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016.

[23] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian. Pose-driven deep convolutional model for person re-identification. In *ICCV*, 2017.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[25] V. Vapnik and R. Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *JMLR*, 2015.

[26] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, pages 791–808, 2016.

[27] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 2017.

[28] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016.

[29] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, pages 2718–2726, 2016.

[30] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

[31] H. Zhao, M. Tian, S. Sun, J. S. andJunjie Yan andShuai Yi, XiaogangWang, and X. Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017.

[32] L. Zhao, X. Li, Y. Zhuang, and JingdongWang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, 2017.

[33] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

[34] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In *ICCV*, 2017.

[35] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, 2017.