Variational Information Distillation for Knowledge Transfer

Sungsoo Ahn [*]
Korea Advanced Institute of Science and Technology
Daejeon, Korea
sungsoo.ahn@kaist.ac.kr

Shell Xu Hu [*]
École des Ponts ParisTech
Champs-sur-Marne, France
hus@imagine.enpc.fr

Andreas Damianou
Amazon
Cambridge, United Kingdom
damianou@amazon.com

Neil D. Lawrence
Amazon
Cambridge, United Kingdom
lawrennd@amazon.com

Zhenwen Dai
Amazon
Cambridge, United Kingdom
zhenwend@amazon.com

Abstract

Transferring knowledge from a teacher neural network pretrained on the same or a similar task to a student neural network can significantly improve the performance of the student neural network. Existing knowledge transfer approaches match the activations or the corresponding hand-crafted features of the teacher and the student networks. We propose an information-theoretic framework for knowledge transfer which formulates knowledge transfer as maximizing the mutual information between the teacher and the student networks. We compare our method with existing knowledge transfer methods on both knowledge distillation and transfer learning tasks and show that our method consistently outperforms existing methods. We further demonstrate the strength of our method on knowledge transfer across heterogeneous network architectures by transferring knowledge from a convolutional neural network (CNN) to a multi-layer perceptron (MLP) on CIFAR-10. The resulting MLP significantly outperforms the-state-of-the-art methods and it achieves similar performance to the CNN with a single convolutional layer.

1. Introduction

Deep neural networks (DNNs) play important roles in various computer vision tasks, *e.g.*, depth estimation [8], pose estimation [26], optical flow [7], object classification [11], detection [10], and segmentation [25]. A typical DNN approach for a computer vision task is to train a sophisticated end-to-end neural network with a large amount of labeled data. Such an approach often delivers state-of-the-art performance if a sufficient amount of data is available. However, in many cases, it is impossible to gather sufficiently large data to train a DNN. For example, in many medical image applications [24], the amount of available data is constrained by the number of patients of a particular disease.

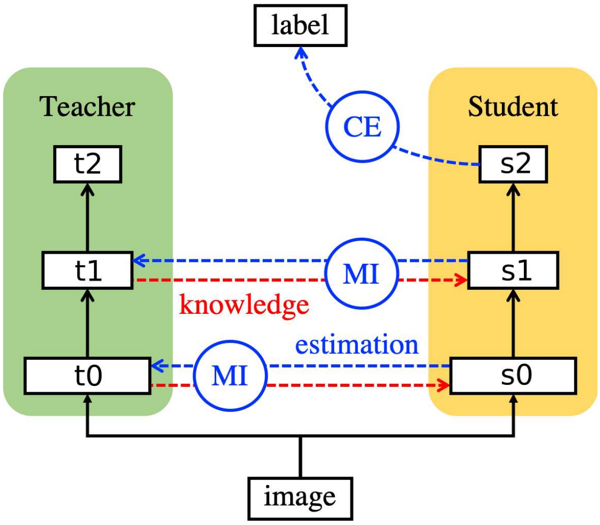


Figure 1: Conceptual diagram of the proposed knowledge transfer method. The student network efficiently learns the target task by minimizing the cross-entropy (CE) loss while retaining high mutual information (MI) with the teacher network. The mutual information is maximized by learning to estimate the distribution of the activations in the teacher network, provoking the transfer of knowledge.

A popular approach for handling such lack of data is transfer learning [19], where the goal is to transfer knowledge from the source task to facilitate learning on the target task. Typically, one considers the source task to be

[*]Contributed during an internship at Amazon.

generic with a larger amount of available data that contains useful knowledge for learning the target task, *e.g.*, knowledge from natural image classification [23] is likely to be useful for fine-grained bird classification [29]. Hinton *et al.* [12] proposed the teacher-student framework for transferring such knowledge between DNNs being trained on the source and target tasks respectively. The high-level idea is to introduce an additional regularization for the DNN being trained on the target task, *i.e.*, the student network, which allows learning the knowledge existing in the DNN that was pre-trained on the source task, *i.e.*, the teacher network. While the framework was originally designed for knowledge transfer between DNNs on the same dataset, recent works [30, 31] started exploiting its potential for more general transfer learning tasks, *i.e.*, when the source data and the target data are different.

Many knowledge transfer methods have been proposed with various intuitions. Hinton *et al.* [12] and Ba and Caruana [2] propose to match the final layers of the teacher and the student network, as the outputs from the final layer of the teacher network provide more information than raw labels. Romero *et al.* [22] proposes to match intermediate layers of the student network to the corresponding layers of the teacher network. Recent works [3, 6, 13, 30, 31] relax the regularization of matching the entire layer by matching carefully designed features/statistics extracted from intermediate layers of the teacher and the student networks, *e.g.*, attention maps [31] and maximum mean discrepancy [13].

Evidently, there is no commonly agreed theory behind knowledge transfer. This causes difficulty in understanding empirical results and in developing new methods in a more principled way. In this paper, we propose variational information distillation (VID) as an attempt towards this direction in which we formulate the knowledge transfer as maximization of the mutual information between the teacher and the student networks. This framework proposes an actionable objective for knowledge transfer and allows us to quantify the amount of information that is transferred from a teacher network to a student network. Since the mutual information is computationally intractable, we employ a variational information maximization [1] scheme to maximize the variational lower bound instead. See Figure 1 for the conceptual diagram of the proposed knowledge transfer method. We further show that several existing knowledge transfer methods [16, 22] can be derived as specific implementations of our framework by choosing different forms of the variational lower bound. We empirically validate the VID framework, which significantly outperforms existing methods. We observe the gap is especially large in the cases of small data and heterogeneous architectures.

In summary, the overall contributions of our paper are as follows:

- We propose variational information distillation, a principled knowledge transfer framework through maximizing mutual information between two networks based on the variational information maximization technique.
- We demonstrate that VID generalizes several existing knowledge transfer methods. In addition, our implementation of the framework empirically outperforms state-of-the-art knowledge transfer methods on various knowledge transfer experiments, including knowledge transfer between (heterogeneous) DNNs on the same dataset or on different datasets.
- Finally, we demonstrate that heterogeneous knowledge transfer between a convolutional neural networks (CNN) and a multilayer perceptrons (MLP) is possible on CIFAR-10. Our method yields a student MLP that significantly outperforms the best-reported MLPs [17, 27] in the literature.

2. Variational information distillation (VID)

In this section, we describe VID as a general framework for knowledge transfer in the teacher-student framework. Specifically, consider training a student neural network on a target task, given another teacher neural network pre-trained on a similar (or related) source task. Note that the source task and the target task could be the same, *e.g.*, for model compression or knowledge distillation. The underlying assumption is that the layers in the teacher network have been trained to represent certain attributes of given inputs that exist in both the source task and the target task. For a successful knowledge transfer, the student network must learn how to incorporate the knowledge of such attributes to its own learning.

From a perspective of information theory, knowledge transfer can be expressed as retaining high mutual information between the layers of the teacher and the student networks. More specifically, consider an input random variable $\boldsymbol{x}$ drawn from the target data distribution $p(\boldsymbol{x})$ and K pairs of layers $\mathcal{R} = \{(\mathcal{T}^{(k)}, \mathcal{S}^{(k)})\}_{k=1}^{K}$, where each pair $(\mathcal{T}^{(k)}, \mathcal{S}^{(k)})$ is selected from the teacher network and the student network respectively. Feedforwarding the input $\boldsymbol{x}$ through the networks induces K pairs of random variables $\{(\boldsymbol{t}^{(k)}, \boldsymbol{s}^{(k)})\}_{k=1}^{K}$ which indicate activations of the selected layers, *e.g.*, $\boldsymbol{t}^{(k)} = \mathcal{T}^{(k)}(\boldsymbol{x})$. The mutual information between the pair of random variables $(\boldsymbol{t}, \boldsymbol{s})$ is defined by:

$$\begin{aligned} I(\boldsymbol{t};\boldsymbol{s}) &= H(\boldsymbol{t}) - H(\boldsymbol{t}|\boldsymbol{s}) \\ &= -\mathbb{E}_{\boldsymbol{t}}[\log p(\boldsymbol{t})] + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log p(\boldsymbol{t}|\boldsymbol{s})], \end{aligned} \tag{1}$$

where the entropy $H(\boldsymbol{t})$ and the conditional entropy $H(\boldsymbol{t}|\boldsymbol{s})$ are derived from the joint distribution $p(\boldsymbol{t}, \boldsymbol{s})$. Empirically, the joint distribution $p(\boldsymbol{t}, \boldsymbol{s})$ is a result of aggregation over the layers with input $\boldsymbol{x}$ sampled from the input distribution

$p(\boldsymbol{x})$. Intuitively, the definition of $I(\boldsymbol{t};\boldsymbol{s})$ can be understood as a reduction in uncertainty in the knowledge of the teacher encoded in its layer $\boldsymbol{t}$ when the the student layer $\boldsymbol{s}$ is known.

We now define the following loss function which aims to learn a student network for the target task while encouraging high mutual information with the teacher network:

$$\mathcal{L} = \mathcal{L}_{\mathcal{S}} - \sum_{k=1}^{K} \lambda_k I(\boldsymbol{t}^{(k)}, \boldsymbol{s}^{(k)}), \tag{2}$$

where $\mathcal{L}_{\mathcal{S}}$ is the task-specific loss function for the target task and $\lambda_k > 0$ is a hyper-parameter introduced for regularization of the mutual information in each layer. Equation (2) needs to be minimized with respect to the parameters of the student network. However, the minimization is hard since exact computation of the mutual information is intractable. We instead propose a variational lower bound for each mutual information term $I(\boldsymbol{t};\boldsymbol{s})$, in which we define a variational distribution $q(\boldsymbol{t}|\boldsymbol{s})$ that approximates $p(\boldsymbol{t}|\boldsymbol{s})$:

$$\begin{aligned} I(\boldsymbol{t};\boldsymbol{s}) &= H(\boldsymbol{t}) - H(\boldsymbol{t}|\boldsymbol{s}) \\ &= H(\boldsymbol{t}) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log p(\boldsymbol{t}|\boldsymbol{s})] \\ &= H(\boldsymbol{t}) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q(\boldsymbol{t}|\boldsymbol{s})] + \mathbb{E}_{\boldsymbol{s}}[D_{\text{KL}}(p(\boldsymbol{t}|\boldsymbol{s})||q(\boldsymbol{t}|\boldsymbol{s}))] \\ &\geq H(\boldsymbol{t}) + \mathbb{E}_{\boldsymbol{t},\boldsymbol{s}}[\log q(\boldsymbol{t}|\boldsymbol{s})], \end{aligned} \tag{3}$$

where the expectations are over the distribution $p(\boldsymbol{t},\boldsymbol{s})$ and the last inequality is due to the non-negativity of the Kullback-Leiber divergence $D_{\text{KL}}(\cdot)$. This technique is known as the *variational information maximization* [1]. Finally, we obtain VID by applying the variational information maximization to each mutual information term $I(\boldsymbol{t}^{(k)}, \boldsymbol{s}^{(k)})$ in (2), leading to a minimization of the following loss function:

$$\widetilde{\mathcal{L}} = \mathcal{L}_{\mathcal{S}} - \sum_{k=1}^{K} \lambda_k \mathbb{E}_{\boldsymbol{t}^{(k)},\boldsymbol{s}^{(k)}}[\log q(\boldsymbol{t}^{(k)}|\boldsymbol{s}^{(k)})]. \tag{4}$$

The objective $\widetilde{\mathcal{L}}$ is jointly minimized over the parameters of the student network and the variational distribution $q(\boldsymbol{t}|\boldsymbol{s})$. Note that the entropy term $H(\boldsymbol{t})$ has been removed from the equation (3) since it is constant with respect to the parameters to be optimized. Alternatively, one could interpret the objective (4) as jointly training the student network for the target task and maximization of the conditional likelihood to fit the activations of the selected layers from the teacher network. By doing so, the student network obtains the “compressed” knowledge required for recovering activations of the selected layers in the teacher network.

2.1. Algorithm formulation

We further specify our framework by choosing a form made for the variational distribution $q(\boldsymbol{t}|\boldsymbol{s})$. In general, we employ a Gaussian distribution with heteroscedastic mean $\boldsymbol{\mu}(\cdot)$ and homoscedastic variance $\boldsymbol{\sigma}$ as the variational distribution $q(\boldsymbol{t}|\boldsymbol{s})$, *i.e.*, the mean $\boldsymbol{\mu}(\cdot)$ is a function of $\boldsymbol{s}$ and the standard deviation $\boldsymbol{\sigma}$ is not. Next, the parameterization of $\boldsymbol{\mu}(\cdot)$ and $\boldsymbol{\sigma}$ is further specified by the type of layer corresponding to $\boldsymbol{t}$. When $\boldsymbol{t}$ corresponds to intermediate layer of the teacher network with spatial dimensions indicating channel, height and width respectively, *i.e.*, $\boldsymbol{t} \in \mathbb{R}^{C\times H\times W}$, our choice of variational distribution is expressed as follows:

$$\begin{aligned} -\log q(\boldsymbol{t}|\boldsymbol{s}) &= -\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W} \log q(t_{c,h,w}|\boldsymbol{s}) \\ &= \sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W} \log \sigma_c + \frac{(t_{c,h,w} - \mu_{c,h,w}(\boldsymbol{s}))^2}{2\sigma_c^2} + \text{constant}, \end{aligned} \tag{5}$$

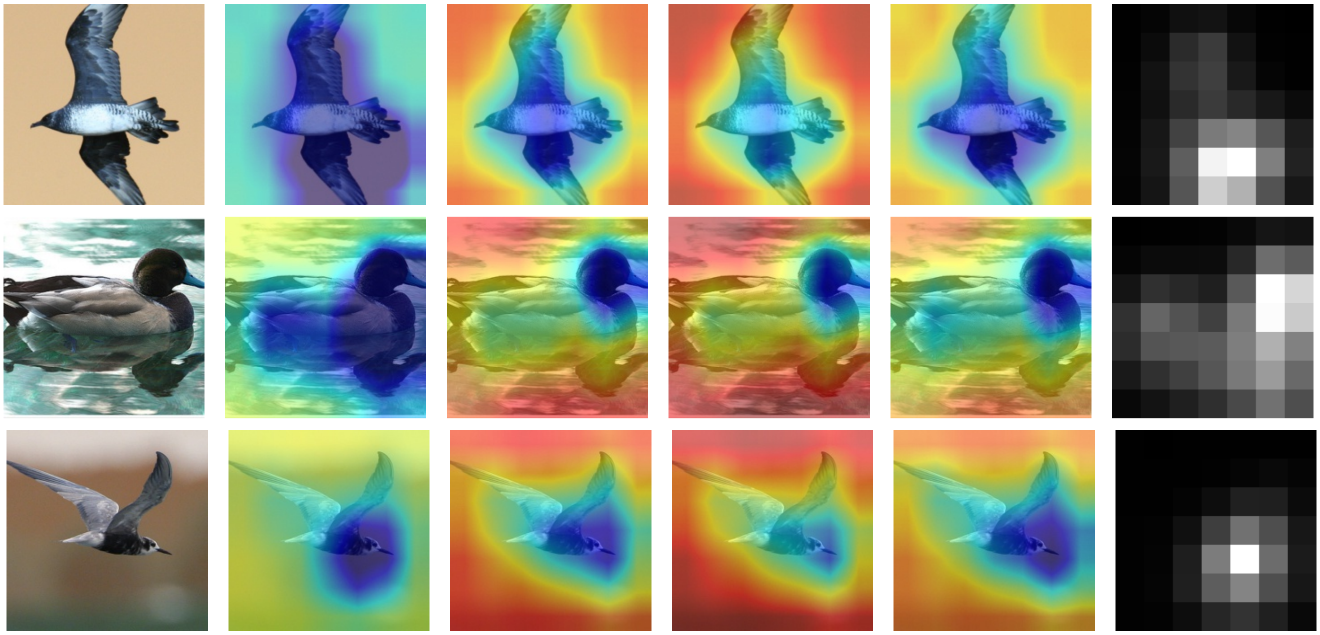
where $t_{c,h,w}$ denote scalar components of $\boldsymbol{t}$ indexed by (c,h,w). Further, $\mu_{c,h,w}$ represents the output of a single unit from the neural network $\boldsymbol{\mu}(\cdot)$ consisting of convolutional layers and the variance is ensured to be positive using the softplus function, *i.e.*, $\sigma_c^2 = \log(1+\exp(\alpha_c)) + \epsilon$ where $\alpha_c \in \mathbb{R}$ being the parameter to be optimized and $\epsilon > 0$ is minimum variance introduced for numerical stability. Typically, one can choose $\boldsymbol{s}$ from the student network with similar hierarchy and spatial dimension as $\boldsymbol{t}$. When spatial dimension of two layers are equal, 1×1 convolutional layers are typically used for efficient parameterization of $\boldsymbol{\mu}(\cdot)$. Otherwise, convolution or transposed convolution with larger kernel size could be used to match the spatial dimensions.

We additionally consider the case when the layer $\boldsymbol{t} = \mathcal{T}^{(\text{logit})}(\boldsymbol{x}) \in \mathbb{R}^N$ corresponds to the logit layer of the teacher network. Here, our choice of the variational distribution is expressed as follows:

$$\begin{aligned} -\log q(\boldsymbol{t}|\boldsymbol{s}) &= -\sum_{n=1}^{N} \log q(t_n|\boldsymbol{s}) \\ &= \sum_{n=1}^{N} \log \sigma_n + \frac{(t_n - \mu_n(\boldsymbol{s}))^2}{2\sigma_n^2} + \text{constant}, \end{aligned} \tag{6}$$

where t_n indicates the n-th entry of the vector $\boldsymbol{t}$, μ_n represents the output of a single unit of neural network $\boldsymbol{\mu}(\cdot)$ and σ_n is, again, parameterized by softplus function to enforce positivity. For this case, the corresponding layer $\boldsymbol{s}$ in the student network is the penultimate layer $\mathcal{S}^{(\text{pen})}$ instead of the logit layer to match the hierarchy of two layers without being too restrictive on the output of the student network. Furthermore, we found that using a simple linear transformation for the parameterization of the mean function was sufficient in practice, *i.e.*, $\boldsymbol{\mu}(\boldsymbol{s}) = \mathbf{W}\boldsymbol{s}$ for some weight matrix $\mathbf{W}$.

The aforementioned implementations turned out to perform satisfactorily during the experiments. We also consid-



(a) input (b) 0-th epoch (c) 40-th epoch (d) 160-th epoch (e) no transfer (f) magnitude of $\boldsymbol{t}_{h,w}$

Figure 2: Plots for the heat maps corresponding to the variational distribution evaluated for spatial dimensions of the intermediate layer in the teacher network, *i.e.*, $\log q(\boldsymbol{t}_{h,w}|\boldsymbol{s}) = \sum_c \log q(t_{c,h,w}|\boldsymbol{s})$. Each figure corresponds to (a) original input image, (b, c, d) log-likelihood $\log q(\boldsymbol{t}_{h,w}|\boldsymbol{s})$ that was normalized and interpolated to fit the spatial dimension of the input image (red pixels correspond to high probability), (d) log-likelihood of variational distribution optimized for the student network trained without any knowledge transfer applied and (f) magnitude of the layer $\boldsymbol{t}$ averaged for each spatial dimensions.

ered using heteroscedastic variance $\boldsymbol{\sigma}(\cdot)$, but it gave unstable training with ignorable improvements. Other types of parameterizations such as a heavy-tailed distribution or the mixture density network [5] could be used to gain additional performance. We leave these ideas for future exploration.

See Figure 2 for an illustration of the training VID using the implementation based on equation (5). Here, we display the change in the evaluated log-likelihood of the variational distribution aggregated over channels, *i.e.*, $\log q(\boldsymbol{t}_{h,w}|\boldsymbol{s}) = \sum_c \log q(t_{c,h,w}|\boldsymbol{s})$, given input $\boldsymbol{x}$ (Figure 2a) throughout the VID training process. One observes that the student network is trained gradually for the variational distribution to estimate the density of the intermediate layer from the teacher network (Figure 2b, 2c and 2d). As a comparison, we also optimize the variational distribution for the student network trained without knowledge transfer, (Figure 2e). For this case, we observe that this particular instance of the variational distribution fails to obtain high log-likelihoods, indicating low mutual information between the teacher and the student networks. Interestingly, the parts that correspond to the background achieve higher magnitudes compared to that of the foreground in general. Our explanation is that the output of layers corresponding to the background that mostly corresponds to zero activations (Figure 2f) and contains less information, being a relatively easier target for maximizing the log-likelihood of the variational distribution.

2.2. Connections to existing works

The infomax principle. We first describe the relationship between our framework and the celebrated *infomax principle* [18] applied to representation learning [28], stating that "good representation" is likely to contain much information in the corresponding input. Especially, such a principle has been successfully applied to semi-supervised learning for neural networks by maximizing the mutual information between the input and output of the intermediate layer as a regularization to learning the target task, *e.g.*, learning to reconstruct input based on autoencoders [21]. Our framework can be viewed similarly as an instance of semi-supervised learning with modification of the infomax principle: layers of the teacher network contain important information for the target task, and a good representation of the student network is likely to retain much of their information. One recovers the traditional semi-supervised learning infomax principle when we set $\boldsymbol{t}^{(k)} = \boldsymbol{x}$ in the equation (2).

Generalizing mean squared error matching. Next, we explain how existing knowledge transfer methods based on mean squared error matching can be seen as a specific instance of the proposed framework. In general, the methods will be induced from the equation (4) by making a specific choice of the layers $\mathcal{R} = \{(\mathcal{T}^{(k)}, \mathcal{S}^{(k)})\}_{k=1}^{K}$ for knowledge transfer and parameterization of heteroscedastic mean $\boldsymbol{\mu}(\cdot)$ in the variational distribution:

$$-\log q(\boldsymbol{t}|\boldsymbol{s}) = \sum_{n=1}^{N} \frac{(t_n - \mu_n(\boldsymbol{s}))^2}{2} + \text{constant}. \qquad (7)$$

Note that Equation (7) corresponds to a Gaussian distribution with unit variance over every dimension of the layer in the teacher network. Ba and Caruana [2] showed that knowledge can be transferred between the teacher and the student networks that were designed for the same task, by matching the output of logit layers $\mathcal{T}^{(\text{logit})}, \mathcal{S}^{(\text{logit})}$ from the teacher and the student networks with respect to mean squared error. Such a formulation is induced from the equation (7) by letting $\mathcal{R} = \{(\mathcal{T}^{(\text{logit})}, \mathcal{S}^{(\text{logit})})\}$, and $\boldsymbol{\mu}(\boldsymbol{s}) = \boldsymbol{s}$ in the equation (7). This was later extended for knowledge transfer between the teacher and the student networks designed for different tasks by Li and Hoiem [16], through adding an additional linear layer on top of the penultimate layer $\mathcal{S}^{(\text{pen})}$ in the student network to matching with logit layer $\mathcal{T}^{(\text{logit})}$ in the teacher network. This is induced similarly from the equation (7) by letting $\mathcal{R} = \{(\mathcal{T}^{(\text{logit})}, \mathcal{S}^{(\text{pen})})\}$, and $\boldsymbol{\mu}(\cdot)$ being a linear transformation, *i.e.*, $\boldsymbol{\mu}(\boldsymbol{s}) = \mathbf{W}\boldsymbol{s}$. Next, Romero *et al.* [22] proposed a knowledge transfer loss for minimizing the mean squared error between intermediate layers from the teacher and the student networks, with additional convolutional layer introduced for adapting different dimension size between each pair of matched layers. This is recovered from the regularization term in the equation (7) by choosing layers for the knowledge transfer to be intermediate layers of the teacher and the student networks, and $\boldsymbol{\mu}(\cdot)$ being a linear transformation corresponding to a single 1×1 convolutional layer.

These methods are all similar to our implementation of the framework in that they all use Gaussian distribution as the variational distribution. However, our method differs in two key ways: (a) allowing the use of a more flexible nonlinear functions for heteroscedastic mean and (b) modeling different variances for each dimension in the variational distribution. This allows transferring mutual information in a more flexible manner without wasting model capacity. Especially, modeling unit variance for all dimensions of the layer $\boldsymbol{t}$ in the teacher network could be highly restrictive for the student network. To illustrate, the layer of the teacher network might include an activation t_n that contains information irrelevant to the task of the student network, yet requires much capacity for regression of $\mu_n(\boldsymbol{s})$ to t_n. This would raise over-regularization issues, *i.e.*, wasting the majority of the student network's capacity on trying to fit such a unit. Instead, modeling high homoscedastic variance σ_n for such dimension make its contribution ignorable to the overall loss, allowing one to "filter" out such unit in an efficient way.

Comparison with feature matching. Besides the knowledge transfer methods based on mean squared error matching, several works [6, 13, 30, 31] have proposed indirectly matching the handcrafted features extracted from intermediate layers. More specifically, Zagoruyko and Komodakis [31] proposed matching the "attention maps" generated from activations from the layers. Huang and Wang [13] later generalized the attention map to matching the maximum mean discrepancy of the activations. Yim *et al.* [30] proposed matching the feature called the Flow of Solution Procedure (FSP) defined by the Gram matrix of layers adjacent in the same network. Chen *et al.* [6] considered matching the reconstructed input image from the intermediate layers of the teacher and the student networks. These methods could be seen as smartly avoiding the aforementioned over-regularization issue by filtering out information in the teacher network using expert knowledge. However, such methods potentially lead to suboptimal results when the feature extraction method is not apt for the particular knowledge transfer task and may discard important information from the layer of the teacher network in an irreversible way.

3. Experiments

We demonstrate the performance of the proposed knowledge transfer framework by comparing VID to state-of-the-art knowledge transfer methods on image classification. We apply VID to two different locations: (a) VID between intermediate layers of the teacher and the student network (VID-I) and (b) VID between the logit layer of the teacher network and the penultimate layer of the student network (VID-LP). For comparison, we consider the following knowledge transfer methods: the original knowledge distillation (KD) [12], learning without forgetting (LwF) [16], hint based transfer (FitNet) [31], activation-based attention transfer (AT) [31] and polynomial kernel-based neural selectivity transfer (NST) [13]. Note that we consider FitNet as a regularization for training the student network [31] instead of a stage-wise training procedure as first proposed in [22]. We compare knowledge transfer methods for knowledge transfer between same and different datasets, which is commonly referred to as the knowledge distillation and transfer learning tasks respectively.

In all the experiments, we select the same pairs of intermediate layers for knowledge transfer based on VID-I, FitNet, AT and NST. Similarly, the same pairs of layers for knowledge transfer are used for LwF and VID-LP. All the hyper-parameters of all the methods are chosen according to the performance on a validation set, which is 20% of

M	5000	1000	500	100
Teacher	94.26	-	-	-
Student	90.72	84.67	79.63	58.84
KD	91.27	86.11	82.23	64.24
FitNet	90.64	84.78	80.73	68.90
AT	91.60	87.26	84.94	73.40
NST	91.16	86.55	82.61	64.53
VID-I	**91.85**	**89.73**	**88.09**	**81.59**
KD + AT	91.81	87.34	85.01	76.29
KD + VID-I	91.7	88.59	86.53	78.48

Table 1: Experimental results (test accuracy) of knowledge distillation on the CIFAR-10 dataset from teacher network (WRN-40-2) to student network (WRN-16-1) with varying number of data points per class (denoted by M).

(d, w)	(40,2)	(16, 2)	(40, 1)	(16, 1)
Teacher	74.16	-	-	-
Student	74.34	70.42	68.79	65.46
KD	75.80	72.87	70.99	66.03
FitNet	74.29	70.89	68.66	65.38
AT	74.76	71.06	69.85	65.31
NST	74.81	71.19	68.00	64.95
VID-I	75.25	73.31	71.51	66.32
KD + AT	75.86	73.13	71.4	67.07
KD + VID-I	**76.11**	**73.69**	**72.16**	**67.19**

Table 2: Experimental results (test accuracy) of knowledge distillation on the CIFAR-100 dataset from the teacher network (WRN-40-2) to the student networks (WRN-d-w) with varying factor of depth d and width w.

the training set. We carefully pick the set of candidate values of hyper-parameters such that all the values proposed in the original works are included. The presented performances are the average of three repeated runs. More details about experiments are included in the supplementary material. The implementation of the algorithm will be made publicly available shortly.

3.1. Knowledge distillation

We first compare knowledge transfer methods on the traditional knowledge distillation task, where a student network is trained on the same task as the teacher network. By distilling the knowledge from a large teacher network into a small student network, we can speed up the computation for prediction. We further investigate two problems for this task: whether we can benefit from knowledge transfer in the small data regime and how much performance we lose by reducing the size of the student network? Note that we do not evaluate the performance of VID-LP and LwF as they are designed for transfer learning. When applied, KD, VID-LP and LwF delivered similar performance.

Reducing training data. Knowledge transfer can be a computationally expensive task. Given a pre-trained teacher network on the whole training data set, we explore the possibility of using a small portion of the training set for knowledge transfer. We demonstrate the effect of a reduced training set by applying knowledge distillation on CIFAR-10 [15] with four different sizes of training data. We employ wide residual networks (WRN) [15] for the teacher network (WRN-40-2) and the student network (WRN-16-1), where the teacher network is pre-trained on the whole training set of CIFAR-10. Knowledge distillation is applied to four different sizes of training set: 5000 (the full size), 1000, 500, 100 data points per class.

We compare VID-I with KD, FitNet, AT and NST. We also provide performances of the teacher network (Teacher) and the student network trained without any knowledge transfer (Student) as baselines. We choose four pairs of intermediate layers similarly to [31], each of which is located at the end of a group of residual blocks. We implemented VID-I using three 1×1 convolutional layers with hidden channel size as twice of the output channel size. The results are shown in Table 1. Our method, VID-I, outperforms other knowledge transfer methods consistently across all regimes. The performance gap increases as the size of dataset get smaller, *e.g.*, VID-I only drops 10.26% of accuracy even when 100 data points per each class are provided to the student network. There is a 31.88% drop without knowledge transfer and a 15.52% drop for the best baseline, *i.e.*, KD + AT.

Varying the size of the student network. The size of the student network gives a trade-off between the speed and the performance in knowledge transfer. We evaluate the performance of knowledge transfer methods on different sizes of the student network. The teacher network (WRN-40-2) is pre-trained on the whole training set of CIFAR-100. A student network with four choices of size, *i.e.*, WRN-40-2, WRN-16-2, WRN-40-1, WRN-16-1, is trained on the whole training set of CIFAR-100. We compare our VID-I with KD, FitNet, AT and NST along with the Teacher and Student baselines. The choices of intermediate layers are the same as the previous experiment.

The results are shown in in Table 1. As also noticed by Furlanello *et al.* [9], the student network with the same size as the teacher network outperforms the teacher network with all the knowledge transfer methods. One observes that VID-I consistently outperforms FitNet, AT and NST, which correspond to the same choice of layers for knowledge transfer. It also outperforms KD except for the case when the structure of the student network is identical to that of the teacher network, *i.e.*, WRN-40-2, where two methods can be combined to yield the best performance.

3.2. Transfer learning

We evaluate knowledge transfer methods on transfer learning. The teacher network is a residual network (ResNet-34) [11] pre-trained on the ImageNet dataset [23]. We apply transfer learning to improve the performance of two separate image classification tasks. The first task is a fine-grained bird species classification based on the CUB-200-2011 dataset [29], which contains 11,788 images in total for 200 bird species. The second task is an indoor scene classification based on the MIT-67 dataset [20], which contains 15,620 images for 67 classes of indoor scenes. For both tasks, there are a relatively few images per class, which can significantly benefit from knowledge transfer from the ImageNet classification task. To evaluate the performance at various levels of data scarcity, we subsample both datasets into three different sizes (50, 25, 10 per class for MIT-67 and 20, 10, 5 per class for CUB-200-2011) and compare the knowledge transfer methods.

We evaluate the knowledge transfer methods in two scenarios: a smaller student network of the same architecture (ResNet-18) and different architecture (VGG-9) [25]. We compare our VID-I and VID-LP with LwF, FitNet, AT and NST. We evaluate the performance of the student network without transfer learning (Student) as a baseline. For the teacher and the student network with ResNet architecture, we choose the outputs of the third and fourth groups of residual blocks (from the input) as the intermediate layers for knowledge transfer. In the case of the VGG-9 student network, we choose the fourth and fifth max-pooling layers as the intermediate layers for knowledge transfer, which corresponds to the same spatial dimension as the intermediate layers selected from the teacher network. For applying VID-I to the ResNet-18 student network, we use two 1×1 convolutional layers with the size of intermediate channels as half of the output channel size. When the student network is VGG-9, a single 1×1 convolutional layer without non-linearity is used.

The results are shown in Table 3. The knowledge transfer from ResNet-34 to VGG-9 gives very similar performance to the transfer from ResNet-34 to ResNet-18 for all the knowledge transfer methods. This shows that knowledge transfer methods are robust against small architecture changes. Our methods outperform other knowledge transfer methods in all regions of comparison. Both VID-I and VID-LP outperforms baselines that correspond to the same choice of layers for knowledge transfer. For the MIT-67 dataset, we observe that our algorithm outperforms even the finetuning method, which requires pre-training of the student network on the source task.

3.3. Knowledge transfer from CNN to MLP

The transfer learning experiments show the robustness of the knowledge transfer method against small architecture changes. This leads to an interesting question: whether a knowledge transfer method can work between two completely different network architectures. A solution to this question can open a new direction of knowledge transfer and potentially offer solutions to many problems, *e.g.*, speeding up prediction of recurrent neural networks (RNNs) by transferring knowledge from a RNN to a CNN, speeding up prediction of CNN on CPU or low-energy device by transferring knowledge from a CNN to a multi-layer perceptron (MLP).

In this paper, we evaluate the performance of knowledge transfer from CNN to MLP on CIFAR-10. There is a well-known performance gap between CNN and MLP on CIFAR-10 [17, 27]. The state-of-the-art performance on CIFAR-10 with MLP is 78.62% with initialization from auto-encoders [17] and 74.32% using knowledge distillation [27]. Urban *et al.* [27] also trained a single convolutional layer achieving the performance of 84.6% using knowledge distillation.

We apply the knowledge transfer methods in the knowledge distillation setting as mentioned in Section 3.1. We use a teacher network with convolutional layers (WRN-40-2) pre-trained on CIFAR-10. We use a MLP with five fully connected hidden layers as the student network, constructed by stacking one linear layer, three bottleneck linear layers and one linear layer in sequence. Each is followed by a non-linearity activation in between. Here, the bottleneck layer indicates a composition of two linear layers without non-linearity that is introduced to speed up learning by reducing the number of parameters. All the hidden layers have the same h number of units and the bottleneck linear layer is composed of two linear layers with a size of $h \times \frac{h}{4}$ and $\frac{h}{4} \times h$.

The knowledge transfer between intermediate layers is defined between the outputs of four residual groups of the teacher network and the outputs of the first four fully connected layers of the student network. We compare VID-I with KD and FitNet since these knowledge transfer methods do not rely on spatial structures. For the same reason, AT and NST are not applicable to multilayer perceptrons. VID-I is implemented with multiple transposed convolutional layers without non-linearities. Specifically, the inputs for the variational distributions, *i.e.*, the hidden layers of the MLP are treated as a tensor with 1×1 spatial dimensions. Single transposed convolutional layer with a 4×4 kernel, unit stride and zero padding is followed by multiple transposed convolutional layers with a 4×4 kernel, two strides, and single padding to match the spatial dimension of the corresponding layer of the teacher network for knowledge transfer. More details on implementations of the student

M	≈80	50	25	10
Student	48.13	37.69	27.01	14.25
fine-tuning	70.97	66.04	58.13	47.91
LwF	63.43	51.79	41.04	22.76
FitNet	71.34	60.45	54.78	36.94
AT	58.21	48.66	43.66	27.01
NST	55.52	46.34	33.21	20.82
VID-LP	67.91	58.51	47.09	31.94
VID-I	71.34	63.66	60.07	**50.97**
LwF + FitNet	70.97	60.37	54.48	38.73
VID-LP + VID-I	**71.87**	**65.75**	**61.79**	50.37

(a) MIT-67, ResNet-34 to ResNet-18

M	≈80	50	25	10
Student	53.58	43.96	29.70	15.97
fine-tuning	65.97	58.51	51.72	39.63
LwF	60.90	52.01	41.57	27.76
FitNet	70.90	64.70	54.48	40.82
AT	60.90	52.16	42.76	25.60
NST	55.60	46.04	35.22	21.64
VID-LP	68.88	61.64	50.22	39.25
VID-I	**72.01**	**67.01**	**59.33**	**45.90**
LwF + FitNet	70.52	64.10	54.63	40.15
VID-LP + VID-I	71.72	66.49	58.96	45.89

(b) MIT-67, ResNet-34 to VGG-9

M	≈29.95	20	10	5
Student	37.22	24.33	12.00	7.09
fine-tuning	76.69	71.00	59.25	44.07
LwF	55.18	42.13	26.23	14.27
FitNet	66.63	56.63	46.68	31.04
AT	54.62	41.44	28.90	16.55
NST	55.01	41.87	23.76	15.63
VID-LP	65.59	54.12	39.20	27.86
VID-I	**73.25**	**67.20**	**56.86**	**46.21**
LwF + FitNet	68.69	58.81	48.86	31.30
VID-LP + VID-I	69.71	63.94	52.87	41.12

(c) CUB-200-2011, ResNet-34 to ResNet-18

M	≈29.95	20	10	5
Student	44.59	32.10	15.69	9.66
fine-tuning	60.96	51.86	46.88	39.98
LwF	52.18	38.05	25.57	13.93
FitNet	68.96	61.52	48.04	32.89
AT	56.28	43.96	28.33	13.98
NST	56.55	44.95	28.43	14.66
VID-LP	66.82	55.94	38.10	30.47
VID-I	**71.51**	**65.69**	53.29	38.09
LwF + FitNet	70.56	62.44	47.36	30.52
VID-LP + VID-I	70.00	65.14	**53.78**	**38.76**

(d) CUB-200-2011, ResNet-34 to VGG-9

Table 3: Experimental results (test accuracy) of transfer learning from the teacher network (ResNet-34) to the student network (ResNet-18/VGG-9) for the MIT-67/CUB-200-2011 dataset with varying number of data points per class (denoted by M). We use $M \approx M_{\text{avg}}$ to denote the setting where the number of data points per class is non-uniform and M_{avg} in average. Fine-tuning gives good results on transfer learning, but is not directly comparable as it is not a knowledge transfer method.

Network	MLP-4096	MLP-2048	MLP-1024
Student	70.60	70.78	70.90
KD	70.42	70.53	70.79
FitNet	76.02	74.08	72.91
VID-I	**85.18**	**83.47**	**78.57**
Urban *et al.* [27]	74.32		
Lin *et al.* [17]	78.62		

Table 4: Experimental result (test accuracy) of distillation on CIFAR-10 from the convolutional teacher network (WRN-40-2) to the fully connected student network (MLP-h) with varying size of hidden dimensions h.

network and the auxiliary distribution are in the supplementary material.

The results are shown in Table 4. Both FitNet and VID-I improve the performance comparing the baseline of directly training the intermediate layers of the student network. VID-I significantly outperforms FitNet on MLPs with different sizes. Furthermore, MLP-4096 outperforms the the state-of-the-art performance with MLP reported by Lin *et al.* [17] (78.62%) and Ba *et al.* [27] (74.32%) significantly. More importantly, our method bridges the performance gap between CNN (84.6% using one convolutional layer [27]) and MLP shown in previous works.

4. Conclusion

In this work, we proposed the VID framework for effective knowledge transfer by maximizing the variational lower bound of the mutual information between two neural networks. The implementation of our algorithm is based on Gaussian observation models and is empirically shown to outperform other benchmarks in the distillation and transfer learning tasks. Using more flexible recognition models, *e.g.*, [14], for accurate maximization of mutual information and alternative estimation of mutual information, *e.g.*, [4], are both ideas of future interest.

References

[1] D. B. F. Agakov. The IM algorithm: a variational approach to information maximization. 2004.

[2] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[3] V. Belagiannis, A. Farshad, and F. Galasso. Adversarial network compression. In *European Conference on Computer Vision*, pages 431–449. Springer, 2018.

[4] I. Belghazi, S. Rajeswar, A. Baratin, R. D. Hjelm, and A. Courville. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[5] C. M. Bishop. Mixture density networks. Technical report, Citeseer, 1994.

[6] S. Chen, C. Zhang, and M. Dong. Coupled end-to-end transfer learning with generalized Fisher information. In *Computer Vision and Pattern Recognition*, 2018.

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[9] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar. Born again neural networks. In *ICML*, 2018.

[10] R. Girshick. Fast r-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[13] Z. Huang and N. Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

[14] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

[15] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[16] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[17] Z. Lin, R. Memisevic, and K. Konda. How far can we go without convolution: Improving fully-connected networks. *arXiv preprint arXiv:1511.02580*, 2015.

[18] R. Linsker. An application of the principle of maximum information preservation to linear systems. In *Advances in neural information processing systems*, pages 186–194, 1989.

[19] S. J. Pan, Q. Yang, et al. A survey on transfer learning. 2010.

[20] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 413–420. IEEE, 2009.

[21] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[22] 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.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[24] T. Schlegl, J. Ofner, and G. Langs. Unsupervised pre-training across image domains improves lung tissue classification. In *International MICCAI Workshop on Medical Computer Vision*, pages 82–93. Springer, 2014.

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

[26] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[27] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, A. Mohamed, M. Philipose, M. Richardson, and R. Caruana. Do deep convolutional nets really need to be deep and convolutional? In *ICLR*, 2017.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[29] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[30] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning.

[31] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2016.