Learning from Multiple Teacher Networks

Shan You
Key Lab. of Machine Perception (MOE),
Cooperative Medianet Innovation Center,
School of EECS, Peking University
Beijing, China
youshan@pku.edu.cn

Chang Xu
UBTech Sydney AI Institute, School of IT,
FEIT, The University of Sydney
Sydney, Australia
c.xu@sydney.edu.au

Chao Xu
Key Lab. of Machine Perception (MOE),
Cooperative Medianet Innovation Center,
School of EECS, Peking University
Beijing, China
xuchao@cis.pku.edu.cn

Dacheng Tao
UBTech Sydney AI Institute, School of IT,
FEIT, The University of Sydney
Sydney, Australia
dacheng.tao@sydney.edu.au

ABSTRACT

Training thin deep networks following the student-teacher learning paradigm has received intensive attention because of its excellent performance. However, to the best of our knowledge, most existing work mainly considers one single teacher network. In practice, a student may access multiple teachers, and multiple teacher networks together provide comprehensive guidance that is beneficial for training the student network. In this paper, we present a method to train a thin deep network by incorporating multiple teacher networks not only in output layer by averaging the softened outputs (dark knowledge) from different networks, but also in the intermediate layers by imposing a constraint about the dissimilarity among examples. We suggest that the relative dissimilarity between intermediate representations of different examples serves as a more flexible and appropriate guidance from teacher networks. Then triplets are utilized to encourage the consistence of these relative dissimilarity relationships between the student network and teacher networks. Moreover, we leverage a voting strategy to unify multiple relative dissimilarity information provided by multiple teacher networks, which realizes their incorporation in the intermediate layers. Extensive experimental results demonstrated that our method is capable of generating a well-performed student network, with the classification accuracy comparable or even superior to all teacher networks, yet having much fewer parameters and being much faster in running.

CCS CONCEPTS

•Computing methodologies →Transfer learning; Neural networks; *Supervised learning by classification;* **•Mathematics of computing** →Network optimization;

KDD'17, August 13–17, 2017, Halifax, NS, Canada..

DOI: http://dx.doi.org/10.1145/3097983.3098135

KEYWORDS

deep learning; multiple teacher networks; knowledge transfer; triplet loss

1 INTRODUCTION

Nowadays, deep neural networks (DNNs) have been highlighted for their remarkable achievements in many tasks, such as image annotation [37] and object classification [25]. These networks differ in the architecture and structure for distinct ideas of the design. They are often composed of a large volume of parameters with considerable *width* and *depth*, so that the function capacity can be maximized. On the other hand, the sheer size of parameters implies the demanding memory storage and computation resources in the inference stage. Wide and deep networks thus cannot accommodate the most low-end computation devices, such as smartphones and tablets. Moreover, the overmuch parameters may induce the networks prone to overfitting [29, 30]. One natural remedy is to directly train a new *thiner* but deep network with fewer parameters, hoping that it still has the excellent performance. Larger depth size is advocated since it has been demonstrated to induce more abstract and invariant representations at higher layers [2] and further improve the classification accuracy. Nevertheless, *deeper* networks could be rather challenging to train [8, 19], considering the degree of networks' non-convexity and non-linearity would dramatically increase.

Squeezing the wide deep networks at hand into a new thin deep network has received increasing attention in recent years. The well-trained wide deep networks are naturally regarded as teachers, which have the capability of guiding the training of a new student network of the smaller size. The student-teacher learning paradigm is inspired by the learning principle underlying the cognitive process of human learning. The intuition can be explained in analogous to human education in which a student is supposed to understand and learn a concept better provided with instructive comments and guidance from teachers. Many efforts have been made to transfer the valuable knowledge from the teacher networks for the training of the student network, so that the corresponding training cost can be eased. For example, [15] proposed to mimic the softened

output of the teacher network; in this way, the dark knowledge can be distilled into the student network. During training, the student network not only approximates the output of the teacher network, but also attempts to predict the ground-truth labels. Besides the softened output of the teacher network, the property of its intermediate representational layers has been also explored for the knowledge transfer towards the student network. In this way, both the intermediate representational layers and the output layers of the student network will receive the supervision from the teacher network. Fitnets [26] required the output of some intermediate layer of the student network to linearly predict that of a teacher network's intermediate layer. Representational distance learning (RDL) [24] proposed to keep the exact pairwise distance between intermediate representations in both teacher and student networks.

Two heads are better than one. A student, in practice, does not solely learn from a single teacher. Instead, he or she may receive the guidance on learning a knowledge point from different teachers in the school, after the school, or even on the Internet. By doing so, the student can fuse different illustrations from multiple teachers to establish his own comprehensive and in-depth understanding of the knowledge. Equivalently, we can also employ multiple teacher networks to help the learning of student networks. There are already some off-the-shelf networks of carefully tailored architectures and promising performance, such as Alexnet [18], VGG16(19) net [29], Resnet [14] and Googlenet [30]. Although there is an absolute rank in their classification accuracy, we suggest that each network can provide some distinct and useful comments that are beneficial for training the student network. However, most of the current literatures do not focus much on this problem, which to some extent causes the waste of resources. The idea of combining multiple teacher networks is consistent with that of multi-view learning [34–36], which assumes that multiple views are more helpful for the learning task than a single view.

In this paper, we investigate training a thin and deep student network by integrating the knowledge from multiple teacher networks. In specific, we employ the information of the intermediate and output layers in teacher networks to regularize some layers of the student networks. Different from FitNets [26] and RDL [24], we suggest that the relative dissimilarity between feature maps of different examples generated by intermediate layers serves as a more flexible and appropriate guidance from teacher networks. Based on a triplet fashion, the relative dissimilarity of examples is encouraged to be maximally maintained between the student network and the teacher networks, acting exactly as the transferred knowledge. Note that different teacher networks tend to have distinct relative dissimilarity relationships. We leverage a voting strategy to unify multiple relative dissimilarity information, which can be transferred into the student network. We evaluated the proposed algorithm in several benchmark datasets including CIFAR-10, CIFAR-100, MNIST and SVHN. The experimental results demonstrated that our method is capable of generating a well-performed student network, with the classification accuracy comparable or even superior to all teacher networks, yet having much fewer parameters and being much faster in running. Moreover, our method also outperforms the current state-of-the-art methods in student-teacher paradigm, partially validating the effectiveness of the proposed relative dissimilarity transfer mechanism and the benefits from incorporation of multiple teacher networks.

The rest of the paper is organized as follows. In Section 2, we discuss some related work that shares similar motivation or methodologies with our method. Then Section 3 formally illustrates our proposed method, including the usage of the triplet loss for relative dissimilarity guidance and the core unification of the guidance from multiple teacher networks. We conduct various experiments on benchmark datasets to validate the effectiveness of our proposed method in Section 4. Ultimately, concluding remarks are attached in Section 5.

2 RELATED WORK

Our work is related to three main aspects or problems among the literatures, which are model compression, training networks with considerable depth and the student-teacher learning paradigm. We will discuss these three aspects respectively.

2.1 Model compression.

Given the fact that the parameters in the huge networks usually have significant redundancy, a number of model compression methods have been developed to compress the well-trained network at hand for a new fast-to-execute and memory-saving model, which has the same or comparable performance. [7] approximately decomposed the large weight matrices into the multiplication of two smaller matrices and [22] utilized a sparse decomposition to reduce the parameters. Or by pushing some neurons to share the same weights, the number of free parameters can also be largely reduced. [9] utilized vector quantization to make similar weights to share the same cluster center and [5] adopted hashing trick to aggressively assign the co-sharing pattern. Extremely, [6] pushed all weights into binary values (1 & -1) with satisfying performance still achieved. Instead of retaining the original structure of the networks, [13] pruned the unimportant neurons. More recently, some complete compression frameworks have also been developed. For example, deep compression [12] integrated pruning, quantization, and Huffman coding to comprehensively compress the networks without affecting accuracy. CNNPack [32] treated filters as images, and adapted image compression techniques for help. In contrast, the proposed algorithm aims to build a new network, which itself has fewer parameters and faster running speed than that of the reference network, but with comparable performance.

2.2 Training networks with considerable depth.

Training a new network which has fewer parameters and comparable performance with the target network usually needs to increase the depth to accommodate demanding modelling capacity. However, the increasing depth may worsen the vanishing gradient phenomenon and induce the training difficulty as a result. Without the help of a teacher network, some algorithms used pre-training strategy to train the network layer by layer and initialized the network with a potentially good start in supervised manner [3] or unsupervised manner [3, 16]. Other literatures attempted to add auxiliary loss functions in the intermediate layers, in order to enhance the weak learning signal caused by traditional backpropagation. For example, [33] proposed a variety of semi-supervised embeddings

to the intermediate outputs and [11, 20, 30] argued that the discriminability of the internal representations against the labels would make a contribution to the training of deep networks. The supervision is implemented by introducing an MLP and a softmax classifier on the intermediate representations.

2.3 Student-teacher learning paradigm

By regarding the network with large depth but thin width as a student network, off-the-shelf teacher networks can be applied to boost its training process and the resulting performance. This student-teacher learning paradigm has been widely studied, and our proposed method belongs to this learning paradigm as well. It is straightforward to encourage the student network to mimic the outputs of the layer before the softmax layer in the teacher network via mean squared error (MSE) [1, 4]. Dark knowledge distillation (KD) [15] supposed that the student network not only accommodates the true labels, but also captures the structures among labels.

Denote the teacher network as $\mathcal{N}_T$ with parameter θ_T and the student network as $\mathcal{N}_S$ with parameter θ_S. Their pre-softmax outputs (*i.e.* activations) are written as O_T and O_S respectively, namely, $\mathcal{N}_T = \text{softmax}(O_T)$ and $\mathcal{N}_S = \text{softmax}(O_S)$[1]. The student network is regulated to have similar output distribution with that of the teacher network, and its outputs should be close to the ground-truth labels, which usually leads to the following training objective:

$$\mathcal{L}_{KD}(\theta_S) = \mathcal{H}(\mathbf{y}, \mathcal{N}_S) + \alpha \mathcal{H}(\mathcal{N}_T^\tau, \mathcal{N}_S^\tau) \tag{1}$$

where $\mathcal{H}(\cdot,\cdot)$ is the cross-entropy and α is the weight parameter balancing these two terms. Moreover, $\mathbf{y}$ refers to the ground-truth label vector; $\mathcal{N}_T^\tau$ and $\mathcal{N}_S^\tau$ are the *softened* outputs of $\mathcal{N}_T$ and $\mathcal{N}_S$,

$$\mathcal{N}_T^\tau = \text{softmax}\left(\frac{O_T}{\tau}\right), \quad \mathcal{N}_S^\tau = \text{softmax}\left(\frac{O_S}{\tau}\right).$$

where $\tau > 1$ is a temperature parameter for the softening manipulation. Since the original outputs of networks $\mathcal{N}_T$ and $\mathcal{N}_S$ usually are in one hot code fashion, their softened targets are expected to have richer information *w.r.t.* labels [15]. Minimizing the objective Eq.(1) can realize the knowledge transfer or distillation from the teacher network into the student network. Empirical results show promising performance of the student network with similar or slightly greater depth [26]. However, since the dark knowledge only focuses on the transfer of the output layer, when dealing with deeper student networks (*e.g.* twice deeper than the teacher networks), this method still suffers from the difficulty of training.

Knowledge transfer can be implemented at intermediate (*i.e.* hidden) layers as well. FitNets [26] defined hints as the outputs of an intermediate layer of the teacher network, and encouraged an intermediate layer of the student network (*a.k.a. guided* layer) to have the ability to predict the outputs of some intermediate layer of the teacher network (*a.k.a. hint* layer). FitNets uses a regressor to accomplish the prediction, and the corresponding loss is defined as

$$\mathcal{L}_{HT}(\theta_S, \theta_{ht}) = \frac{1}{2}\left\| O_T^{inter}(\theta_T) - r(O_S^{inter}(\theta_S); \theta_{ht}) \right\|_2^2, \tag{2}$$

where O_T^{inter} and O_S^{inter} are the intermediate layer's outputs of the teacher network and the student network respectively; r is a linear regressor with parameter θ_{ht}, *e.g.* a fully connected regressor or a convolutional one. Thanks to introducing the intermediate constraint, FitNets is capable of training a deeper student network, with comparable or even better classification performance obtained.

Representational distance learning (RDL) [24] also investigated the effectiveness of the knowledge transferred from the intermediate layers of a teacher network. More specifically, it enables a student to learn the intermediate representational spaces of a teacher network by turning to representational distance (or dissimilarity) matrices using a portion of training examples. Its goal is to keep the pairwise distance values between the student network and the teacher network at some intermediate layer by minimizing the following loss:

$$\mathcal{L}_{RDL}(\theta_S) = \frac{1}{2(n^2-n)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} ((d_S^{inter})_{ij} - (d_T^{inter})_{ij})^2 \tag{3}$$

with the pairwise distance defined as

$$(d_S^{inter})_{ij} = d(O_S^{inter}(\mathbf{x}_i), O_S^{inter}(\mathbf{x}_j))$$
$$(d_T^{inter})_{ij} = d(O_T^{inter}(\mathbf{x}_i), O_T^{inter}(\mathbf{x}_j))$$

where $d(\cdot,\cdot)$ is a distance or dissimilarity metric. Our work also draw support from the concept of dissimilarity among examples, however, we put emphasis on the *relative* dissimilarity instead of the absolute value of dissimilarity. Thus our method enjoys more flexibility and relaxation than RDL.

As illustrated previously, adding the auxiliary constraint on the intermediate layers of the student network is regarded as an efficient way to transfer the teacher network's knowledge into the student network. The advantages brought by this practice are mainly two-fold. First, this very constraint can provide guidance during the training to pull the student network into a good local minima. As a consequence, the student network is expected to obtain a better classification performance than that only using the label supervision at the output layer. Second, since the student network can be much deeper than the teacher network, its training may well be trapped in the *vanishing* gradient dilemma. Thus the auxiliary constraint can serve as *stimulations* for the gradients, and lead to good convergence behavior.

3 METHOD

In this section, we elaborate the proposed method, which is capable of incorporating multiple teacher networks to help training a thin and deep student network. In specific, our proposed method can implement the incorporation not only in the output layer but also in the intermediate layers. Formally, given a student network $\mathcal{N}_S$ parameterized by θ_S and a training dataset $\mathcal{D} := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, our goal is to regularize its both output layer and intermediate layers using the knowledge transferred from multiple teacher networks $\{\mathcal{N}_{T_t}\}_{t=1}^m$ parameterized by θ_{T_t}. In the following, we illustrate the constraints imposed on the output layer and the intermediate layers, respectively. Then we discuss how to combine multiple teacher networks in a unified fashion, together with some necessary details presented.

[1]For simplicity, we also denote the output of teacher network and student network as $\mathcal{N}_T$ and $\mathcal{N}_S$ respectively.

As previously illustrated, dark knowledge distillation acts as an effective way to explore the beneficial information of the teacher network's output layer. The student network is encouraged to learn from the softened output of the teacher network via the additional loss $\mathcal{H}(\mathcal{N}_T^\tau, \mathcal{N}_S^\tau)$ in Eq.(1). When dealing with multiple teacher networks, there are also multiple softened outputs. Similar with [15, 26], we choose to average their soft labels, and the corresponding loss is generalized into

$$\mathcal{L}_{KD}^{multi} = \mathcal{H}(\frac{1}{m}\sum_{t=1}^{m}\mathcal{N}_{T_t}^\tau(\mathbf{x}_i), \mathcal{N}_S^\tau(\mathbf{x}_i)). \tag{4}$$

The averaged softened output serves as the incorporation of multiple teacher networks in the output layer. Minimizing the loss Eq.(4) fulfills the knowledge transfer at this layer. Moreover, the averaged softened output is supposed to be more objective than any of individuals, for it can weaken the unexpected bias of the softened output existing in some examples.

However, only with the distilled dark knowledge in the output layer, the student network may still suffer from the difficulty of training for its considerable depth [26]. Similarly, we investigate to incorporate multiple teacher networks in intermediate layers as well. As discussed above, Fitnets employs the hints in the teacher network's intermediate layers, taking the intermediate output also as the target of the student network's intermediate output. RDL advocates maintaining the representational space by keeping the pairwise distances between representations, utilizing the mean squared error (MSE) to measure the discrepancy of dissimilarity between the teacher network and the student network. Both methods convert the teacher network's guidance into auxiliary constraints on the outputs of intermediate layers. However, the proposed constraints are not suitable for the case of multiple teacher networks. On the other hand, FitNets and RDL propose to learn the exact values of intermediate output and pairwise dissimilarity, respectively. Nevertheless, these two objectives may be too strong for the student network to learn, which is usually caused by the inconsistent value dimensions between the student network and teacher networks with different architectures. Due to this harsh guidance, the flexibility of the student network could be crumbled and might be weak to adaptively learn the network as well as following the teacher's guidance. In the following, we propose to adopt a more flexible constraint imposed on the intermediate representations of the student network. And it can be applied naturally to multiple teacher networks.

3.1 Relative dissimilarity among intermediate representations

As discussed in [33], dissimilarity among training examples (or the intermediate representations) can be used to characterize the corresponding representational space. Thus we can take the dissimilarity relationships at intermediate layers as the transferred knowledge to boost the learning of student network. From the point of view in human cognition, it is hard to judge whether two objects A & B are similar or not, together with how much they are similar. However, people are capable of comparing the dissimilarity among more (at least three) objects. For instance, given A&B and A&C, it is easy to judge which pair is more alike or different. This practice is a more flexible and weaker manner to describe the dissimilarity for we do not have to know the exact dissimilarity values but the *relative* rank relationships.

Therefore, we argue the relative dissimilarity among examples in the intermediate representational space is a more flexible and appropriate property within multiple teacher networks. As discussed above, we propose to adopt triplets to describe this very relative dissimilarity. Triplet-based constraint has been verified to have excellent performance in metric learning [17] and classification tasks [27]. In this way, we can encourage the relative similarity relationships in the student network to be consistent with that in teacher networks. By doing this, the knowledge transfer is accomplished in the intermediate layers, and the student network is equipped with sufficient learning flexibility as well. Most notably, this very relative similarity is suitable for the multiple teacher networks, and independent of the various value dimensions existing in different teacher networks. For the simplicity and clarity of the presentation, we first illustrate the case of one single teacher network, and formulate the basic methodology and details. Then we discuss particularly how to combining multiple teacher networks using this relative similarity in the next subsection.

Formally, denote the single teacher network as $\mathcal{N}_T$ parameterized by θ_T. Suppose the index of the target intermediate layer of the student network is id_S, and its output is determined by the parameter w_S, we have $w_S \subset \theta_S$. Thus the intermediate output (*i.e.* learned representations) of each example in id_S layer of student network is calculated as

$$\mathbf{p}_i = \mathcal{N}_S(\mathbf{x}_i; w_S) = \mathcal{N}_S(\mathbf{x}_i; id_S, \theta_S)$$

and that in teacher network is

$$\mathbf{q}_i = \mathcal{N}_T(\mathbf{x}_i; w_T) = \mathcal{N}_T(\mathbf{x}_i; id_T, \theta_T).$$

In this paper, we quantize the dissimilarity between two examples' intermediate output as their distance according to some distance metric $d(\cdot,\cdot)$, *i.e.* $d_S^{ij} = d(\mathbf{p}_i, \mathbf{p}_j)$ and $d_T^{ij} = d(\mathbf{q}_i, \mathbf{q}_j)$. In our method, we utilize the ℓ_2 distance metric. Given a triplet $(\mathbf{p}_i, \mathbf{p}_i^+, \mathbf{p}_i^-)$ where $\mathbf{p}_i$ is the anchor point, there exists a partially ordered relation in terms of the relative similarity, *i.e.* $\mathbf{p}_i^+ \succ_{\mathbf{p}_i} \mathbf{p}_i^-$, which means that $\mathbf{p}_i^+$ is less dissimilar or has smaller distance $d(\mathbf{p}_i, \mathbf{p}_i^+)$ with $\mathbf{p}_i$ than that of $\mathbf{p}_i^-$. Hence, given an arbitrary $(\mathbf{p}_i, \mathbf{p}_{i_1}, \mathbf{p}_{i_2})$, we propose to determine the partially ordered relation according to the teacher network. Specifically, let

$$\mathbf{p}_{i_1} = \begin{cases} \mathbf{p}_i^+, & \text{if } d(\mathbf{q}_i, \mathbf{q}_{i_1}) < d(\mathbf{q}_i, \mathbf{q}_{i_2}) \\ \mathbf{p}_i^-, & \text{otherwise.} \end{cases} \tag{5}$$

Then we keep the relative dissimilarity relationships of the teacher network into the student network, and in this way, we accomplish the knowledge transfer. In specific, we adopt the triplet loss as the loss function to penalize the violation of this relative dissimilarity, and the loss imposed on the intermediate layer of the student network is

$$\mathcal{L}_{RD}(w_S; \mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-) = \max(0, d(\mathbf{p}_i, \mathbf{p}_i^+) - d(\mathbf{p}_i, \mathbf{p}_i^-) + \delta) \tag{6}$$

where $\delta > 0$ is a small number to prevent the trivial solution. From the Eq.(6), we can see when the example $\mathbf{x}_i^+$'s intermediate representation $\mathbf{p}_i^+$ is farther from $\mathbf{p}_i$ than that of example $\mathbf{x}_i^-$'s intermediate representation $\mathbf{p}_i^-$, the ordering given by the teacher

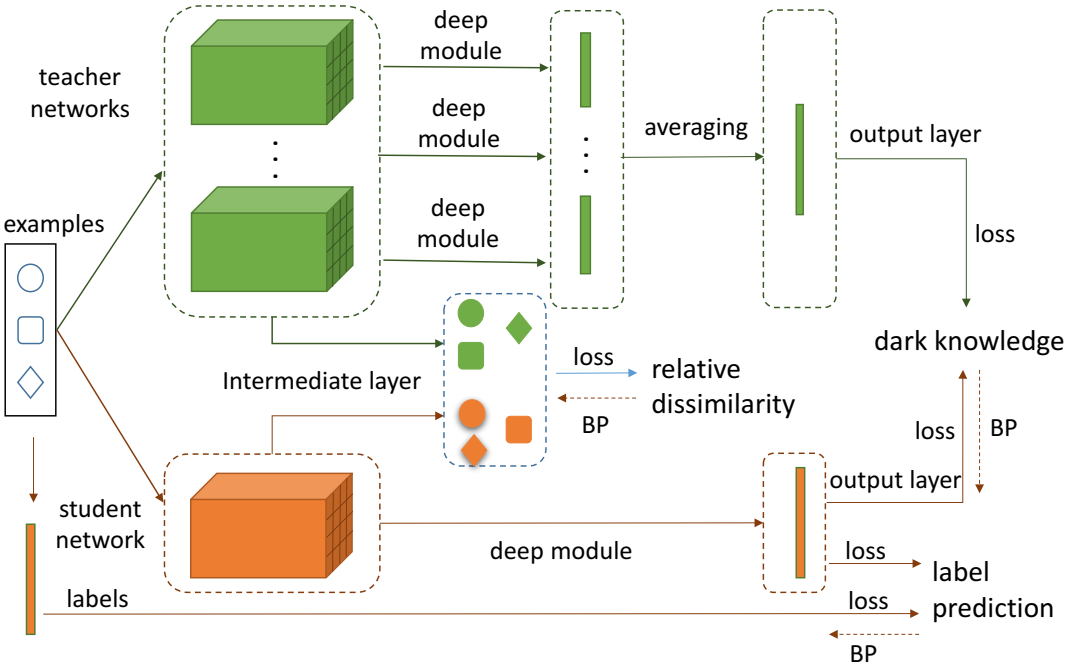


Figure 1: A graphical diagram for the proposed method to train a new thin deep student network by incorporating multiple comparable teacher networks. The method consists of three losses, including label prediction loss, dark knowledge loss and the relative similarity loss. The incorporation of multiple teacher networks exists in two places. One is in the output layers via averaging the softened output targets; the other lies in the intermediate layer by determining the best triplet ordering relationships.

network would be violated and thus a positive loss is obtained. Then minimizing this loss would force the student network to pull the representation $\mathbf{p}_i^+$ back to the anchor $\mathbf{p}_i$ and push away the $\mathbf{p}_i^-$ accordingly. In contrast, when the ordering relationship coheres exactly with that of the teacher network, the loss would be zero and there will be no progress for the training of the student network.

3.2 Combining multiple teacher networks

From Eq.(6) we can see the determination of the triplet's partially ordering relationship is the core of the knowledge transfer from a teacher network into the student network. Multiple teacher networks may have different ordering relationships, thus how to unifying these relationships is the gist of incorporation of multiple teacher networks. Besides, one potential issue is that the feature maps at the convolutional layers can be very noisy [31]. Hence the ordering relationships determined by the teacher network are prone to getting influenced by the annoying noise. Fortunately, when we have multiple comparable teacher networks at hand, this problematic issue can be easily lightened by incorporating them.

Voting strategy. Given m teacher networks $\mathcal{N}_{T_1}, \mathcal{N}_{T_2}, ..., \mathcal{N}_{T_m}$ and a triplet pair $(\mathbf{q}_i, \mathbf{q}_{i_1}, \mathbf{q}_{i_2})$, thus all teacher networks might give two guidance orders, *i.e.* $\mathbf{q}_{i_1} \succ_{\mathbf{q}_i} \mathbf{q}_{i_2}$ or $\mathbf{q}_{i_2} \succ_{\mathbf{q}_i} \mathbf{q}_{i_1}$. Due to the noise or the undulatory property in the representational space, there are usually conflicts of the orders among all teacher networks. To conclude a convincing result of the Oracle order, we can use voting strategy to select the best order; namely, we choose the majority side as the final guidance order suggested by the teacher networks. This practice can not only make the orders more robust in terms of the incident flip of orders due to the noise, but also naturally combine the multiple teacher networks. In this way, the incorporation in the intermediate layer is accomplished.

To fully take advantage of the teacher networks, we integrate the two constraints on both output and intermediate layers, and the whole objective of the student network for the triplet pair $(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)$ and their groundtruth label $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$ can be written as

$$\mathcal{L}(\theta_S) = \sum [\mathcal{H}(\mathbf{y}_i, \mathcal{N}_S(\mathbf{x}_i)) + \alpha \mathcal{H}(\frac{1}{m}\sum_{t=1}^{m} \mathcal{N}_{T_t}^{\tau}(\mathbf{x}_i), \mathcal{N}_S^{\tau}(\mathbf{x}_i))] + \beta \mathcal{L}_{RD}(w_S; \mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-) \quad (7)$$

where $\alpha, \beta \geq 0$ are the weights balancing all three objectives; the $\sum$ sign means the sum over three items in the triplet pairs $(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)$ and $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$. Note that the first term in Eq.(7) refers to the conventional cross-entropy between the output probability of the student network and the corresponding one hot code groundtruth labels; the second term encourages the student network to grasp the fine-grained label information in the output layer learned by the teacher networks; the third term enforces the student network to have the same or similar relative dissimilarity relationships in the intermediate layers as the teacher networks do. The whole method is visualized in Figure.1.

As shown in Figure 1, the student network is provided with three losses, and the whole network's update is implemented by minimizing the loss Eq.(7). The learning process is related with two parameters, *i.e.* α and β, which represent the weight of the output constraint and the intermediate representational constraint, respectively. In our training, we gradually anneal α and β with a linear decay. In this way, the whole process can be regarded as a natural transition from learning under the teachers' guidance to the self-regulated learning.

Note the training process of the student network has the triplet inputs; triplet selection is a key issue in optimizing triplet-based objective losses [27]. In our problem, the triplet loss implies the knowledge distilled from the teacher networks. However, during the network update, if the triplet pair input has the exact relative dissimilarity relationship with that of teacher networks, then its triplet loss would be zero; as a result, this triplet will have no contribution to the update of the student network for its gradient on the triplet loss remains zero. This brings in two disadvantages. On one hand, these *consistent* triplets would *prevent* the student network from being guided by the teacher networks in the intermediate layers for their triplet losses being zero. Then the guidance from teacher networks would be weakened or even crippled, which we do not expect to happen. On the other hand, these non-contributing triplets probably slow the convergence for their corresponding updates with little progress. Thus we tend to select the triplets that violate the triplet constraint (*i.e.* non-zero triplet loss) such that the teacher networks do have a prominent impact on the update of student network. In our method, we propose to generate triplets online [27], namely, selecting the triplets from within a mini-batch that are most inconsistent with the ones provided by teacher networks.

Selecting layers in the student and teacher networks. One open question is that before training, which intermediate layer of the student and teacher networks should be used. As for the selection of the student network's intermediate layer, the middle layer or around empirically tends out to be a good option [26]. As for the selection of intermediate layer for one single teacher network, the middle layer might be also a good choice. However, when it comes to multiple teachers, this problem can be more complicated, and prone to this empirical yet *subjective* determination. Here we present a rule that can perform the layer selection driven by the data tactfully as well as affected by the human empirical decision.

As discussed previously, there exist conflicts about the order of triplet pair $(\mathbf{q}_i, \mathbf{q}_{i_1}, \mathbf{q}_{i_2})$ among all teacher networks. Let $[1:m]$ be the set containing all positive numbers no greater than m. Besides, denote the indexes of all teacher networks satisfying $\mathbf{q}_{i_1} \succ_{\mathbf{q}_i} \mathbf{q}_{i_2}$ as $\phi_i \subset [1:m]$ and $\psi_i \subset [1:m]$ otherwise. Given n triplet pairs $\mathbf{e}_i = (\mathbf{q}_i, \mathbf{q}_{i_1}, \mathbf{q}_{i_2}), i = 1, ..., n$, the *conflicts index* of a specific intermediate layer selection $(id_{T_1}, ..., id_{T_m})$ of teacher networks on these triplets is defined as

$$c(\mathcal{E}; (id_{T_1}, ..., id_{T_m})) = \frac{1}{mn} \sum_{i=1}^{n} \min(|\phi_i|, |\psi_i|). \tag{8}$$

where $\mathcal{E} = \{\mathbf{e}_1, ..., \mathbf{e}_n\}$ is the group of selected triplets.

We can see the conflicts index c reflects for a specific selection of layers in teacher networks, how many networks are inconsistent with the resulting order relationships under the voting strategy. Since different selections of intermediate layers in teacher networks cause different conflicts index, we argue that truly valuable intermediate layers can be reflected in most teacher networks. Thus among these valuable layers, the relative similarity between representations are consistent with each other to the greatest extent, namely, the smallest conflicts index. For employing the human's experience, in practice we can give a candidate selection set ID_{T_i} for each teacher network, then adopt exhaustive search in all possibilities $ID_T = ID_{T_1} \times ID_{T_2} \times ... \times ID_{T_m}$. Finally, the one with smallest conflicts index is determined as the ultimate selection, *i.e.*

$$(id^*_{T_1}, ..., id^*_{T_m}) = \arg \min_{(id_{T_1}, ..., id_{T_m}) \in ID_T} c(\mathcal{E}; (id_{T_1}, ..., id_{T_m})) \tag{9}$$

Note that the exhaustive selection is implemented before the training of the student network, thus the selection complexity does not dominate the entire time complexity. Basically, with the increase of the group $\mathcal{E}$'s size n, the obtained conflict index can reflect better the consistency condition of teacher networks; however, in real implementation, we do not need to enumerate all possible triplets, and a quantity in the same order of training dataset size would empirically suffice for this layer selection task.

There are also some potential extensions for our method, and we discuss them as follows.

Remark 1. When training an extremely deep student network, the constraint on a single intermediate layer may be not enough. Fortunately, our method can be extended into multiple intermediate layers easily. Besides, the layers selection of teacher networks can still follow the criterion Eqs.(8) and (9). In this paper, we only focus on a single guided intermediate layer of student network for simplicity, though we believe that multiple guided layers walk the same steps with the single one.

Remark 2. Like the generalized distillation [23], our method also can be extended to semi-supervised cases. Since the losses related with teacher networks in objective Eq.(7) are both label-free, the numerous unlabeled examples can still be involved in the training. In specific, the teacher networks can prepare soft labels for all the unlabeled examples as well as provide guidance to the intermediate layers of student network. Then the labeled examples are capable of fine tuning the student network to further improve the performance.

4 EXPERIMENT

In this section, we implement experiments to validate the effectiveness of our proposed method on four benchmark datasets, including CIFAR-10, CIFAR-100, MNIST and SVHN. Moreover, we analyze the empirical results in order to further investigate the benefits of our proposed method.

4.1 Validation on CIFAR-10

The CIFAR-10 dataset is composed of 32×32 pixel RGB color images drawn from ten categories. The whole dataset of 60,000 images is split into 50,000 training and 10,000 testing images. In specific, we pre-process these images using global contrast normalization (GCA) and ZCA whitening. And for the selection of hyperparameters in all the following experiments, we take the last 10,000 training examples as the validation ones; the optimal parameter is determined by the top performance on the validation dataset, then we train the

Table 1: Tradeoff of performance in compression rate, acceleration rate and classification accuracy.

	# layers	# params	# mult	accelaration rate	compression rate	Fitnets	Ours
Teachers (top)	5	~9M	~725M	1	1	90.21%	
Student 1	11	~250K	~30M	**×13.17**	**×36**	89.03%	89.37%
Student 2	11	~862K	~108M	×4.56	×10.44	91.01%	91.12%
Student 3	13	~1.6M	~392M	×1.40	×5.62	91.14%	91.25%
Student 4	19	~2.5M	~382M	×1.58	×3.60	91.55%	**91.66%**

ultimate model by the whole training dataset (including validation dataset) under the selected parameter configuration.

As for the teacher networks, we follow the maxout convolutional network presented in [10]. The reported network is composed of 3 convolutional layers of 96-192-192 units respectively, which are also followed by a maxout nonlinearity activations with 2 linear pieces and a maxpooling operator accordingly. Then a fully-connected layer of 500 units with 5-linear-piece maxout activations is built on the top of the convolutional module, and is followed by the final softmax layer. The network was trained by stochastic gradient descent with learning rate decay and momentum. Including this, we also trained two additional teacher networks; one is in the same architecture with it but started with a different initialization, and the other shares the same convolutional kernel size and similar amount of parameters (~9M), but with different units, *i.e.* 96-128-256-384-10.

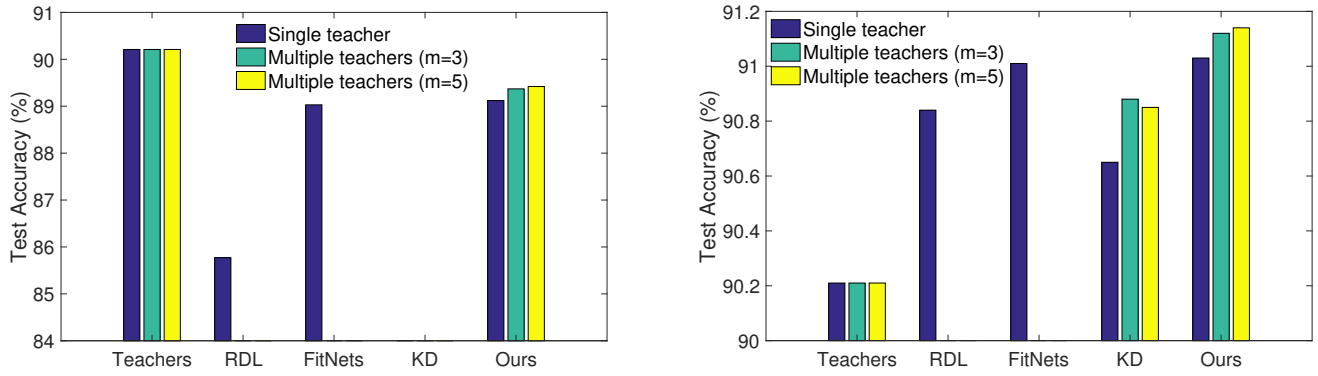
4.1.1 Multiple teacher networks VS single teacher network. We first investigate the progress in performance by incorporating these three (m=3) teacher networks. As for the student networks, we followed the architecture provided in [26] for fair comparison and evaluation, where all student networks are composed of a series of convolutional layers with same convolutional kernel size 3×3, and a fully-connected layers followed by the final softmax layer. All student networks just vary in the number of convolutional layers and the corresponding number of channels. In this experiment, we selected two 11-layer student networks with a different number of parameters and the intermediate loss was empirically imposed on the 6-th layer. As for the determination of three teacher networks, we adopted the rule Eq.(9) in the candidate layer index set $\{2,3\} \times \{2,3\} \times \{2,3\}$. The parameter δ in Eq.(6) was set into 1e-4. The α and β were decayed linearly among epochs and their initial values were determined via monitoring validation set. Moreover, we augmented the dataset with random flipping as [10, 20, 26]. We trained the student networks by transferring the knowledge from all three teacher networks. As for the single-teacher scenario, we choose to learn from the teacher network with top classification accuracy.[2] The results are presented in Figure 2.[3]

Note that only dark knowledge distillation (KD) and ours consider the multiple teacher networks. Comparing both methods in Figure 2 for three teacher networks ($m = 3$), we can see when the student network has a too small capacity, KD method even failed to train it while our method still could achieve comparable performance with that of teacher networks. Increasing the parameters, KD method enabled to obtain a satisfying result, but our method still outperformed it. Besides, we can see the incorporation of multiple teacher networks can significantly improve the performance of one single teacher network (90.65→90.88 for KD and 91.03→91.12 for ours). This progress attributes to the sensible incorporation and more informative knowledge transfer. Note the traditional backpropagation method failed in both cases when training such a deep network with 11 layers, which in a way validates the constraint from output layer and intermediate layer does help the training of DNNs. Considering the single-teacher scenario, our method is slightly better than FitNets and RDL, which implies the effectiveness of our used relative dissimilarity constraint; however, after combining multiple teacher networks, our method is much superior to them and achieves the top accuracy. Then we can safely conclude that our method provides an effective way to incorporate the informative knowledge of multiple teachers in the output and intermediate layers, which the student network benefits from and achieves excellent classification performance as a result.

Then we trained two more teacher networks as previously discussed. The student networks are learned similarly but with five ($m = 5$) teacher networks. The results are also presented in Figure 2. We can see the additional two teacher networks enable the student network learned by our method to improve to some extent. This may result from the voting strategy benefits from more voters (*i.e.* teacher networks). Also note that the student network learned by KD method has a slight decrease in the classification accuracy. It might be because the newly introduced teacher networks do not have an informative softened output layer as the other three ones. Somewhat interfering bias is thus caused in the averaged softened output layer.

4.1.2 Statistics in compression and accelaration. We now evaluate the performance of our method in the application of model compression, especially the tradeoff among compression rate, acceleration rate and classification accuracy. We follow the 4 student networks in [26], which have a different number of layers and parameters. The compression rate of each student network is reflected by its number of parameters. The number of multiplications refers to how many multiplications an example needs during its forward calculation. To obtain the real acceleration rate in practice, we recorded the average time of running the test datasets. All student networks were trained by incorporating the three teacher networks. We also reported the corresponding results of FitNets, where the teacher network was determined to be the one with top performance. The results are summarized in Table 1. Note that the results of "Teachers" in Tabel 1 are referred to the top teacher network. The smallest Student 1 achieves the greatest compression rate ×36 and acceleration rate ×13.17, yet it only dropped 1.31% and 0.93% in the accuracy for FitNets and Ours respectively. Increasing

[2]We call it the *top teacher network* in the following.
[3]The difference between our obtained results and those reported in [10, 26] may be induced by that the script is run on a different machine.



(a) Student: 11 layers & ~250K params

(b) Student: 11 layers & ~862K params

Figure 2: Comparison of single-teacher methods and multiple-teacher methods in training the student networks. Both two student networks have the same 11 layers, but with $\sim 250K$ and $\sim 862K$ parameters, respectively. For the single-teacher scerario, the student networks are learned from the top teacher network, which has 5 layers and roughly 9M parameters. Note that the empty bars of KD in (a) means we can not train a passable student network with such deep layers and small capacity. The results of "Teachers" are referred to the top teacher network.

Table 2: Classification accuracy on CIFAR-10 and CIFAR-100. Note that the student networks for all methods are identical. The results of "Teachers" are referred to the top teacher network.

Algorithm	# layers	# params	CIFAR-10	CIFAR-100
Teachers	5	~9M	90.21%	62.78%
Ours	19	~2.5M	**91.66%**	**65.06%**
Student-teacher learning paradigm				
FitNets [26]			91.55%	64.89%
Knowledge Distillation [15]			91.04%	63.07%
State-of-the-art methods				
Maxout Networks [10]			90.62%	61.43%
Network in Network [21]			91.20%	64.32%
Deeply-Supervised Networks [20]			**91.78%**	**65.43%**

the parameters, Student 2, 3 and 4 begin to outperform the top teacher network, but all of them still remain faster than the top teacher network. Even the deepest Student 4 network achieves the top accuracy with roughly 1/3 of parameters and much better and faster than the teacher networks. Also comparing the FitNets and ours we can find our method has better classification performance in all cases, indicating our method enjoys the benefit from the incorporation of multiple teacher networks.

4.2 Various empirical results

To comprehensively illustrate the effectiveness of our proposed method, including CIFAR-10 we also investigate the performance on the other three datasets, *i.e.* CIFAR-100, MNIST and SVHN. Reported results about multiple teacher networks refer to the top one.

4.2.1 CIFAR-10 & CIFAR-100. The CIFAR-100 dataset has the same size and format with CIFAR-10 dataset, namely, 60,000 color images of pixel 32×32. Nevertheless, CIFAR-100 dataset consists of 100 objects in all, which implies more challenge than CIFAR-10. Similarly, we also pre-process these images using global contrast normalization and ZCA whitening. The three teacher networks and the student network have the same configuration with that of CIFAR-10, except for the number of units in the ultimate softmax layer being 100 instead of 10. We also augmented dataset via random flipping.

Table 2 shows that in both datasets, the student network trained by our method can significantly outperform all teacher networks, but with roughly 1/3 of parameters and 4 times of layers. Besides, our method is much superior to the knowledge distillation method, which implies the advantage of guidance in intermediate layers. Moreover, our method still surpasses other single-teacher algorithms, indicating the benefits from incorporation of multiple teacher networks. When compared to the state-of-the-art methods, our method matches them.

4.2.2 MNIST. The MNIST dataset is composed of 28×28 pixel greyscale images of handwritten digits 0-9, with 60,000 training and 10,000 test examples. Similarly, we follow the maxout convolutional network reported in [10] as a teacher network, which has 3 convolutional maxout layers followed by a fully-connected softmax layer with 48-48-24-10 units respectively. Then we trained two additional teacher networks as CIFAR-10 did. The student network consisted of six convolutional layers and a softmax layer, but with only 8% of parameters. The 4-th layer of the student network acted as the intermediate layer guided by the teacher networks' 2-th layers. Table 3 summarizes the obtained results. It can be concluded that our student network still outperforms all the teacher networks with much fewer parameters. Besides, via incorporating multiple teacher networks, our method achieves better results

Table 3: Classification error on MNIST. Note that the student networks for all methods are identical. The results of "Teachers" are referred to the teacher network with smallest classification error.

Algorithm	# layers	# params	Error
Teachers	4	~361K	0.55%
Ours	7	~30K	**0.48%**
Student-teacher learning paradigm			
FitNets [26]			0.51%
Knowledge Distillation [15]			0.63%
State-of-the-art methods			
Standard backpropogation *w.r.t.* labels			1.90%
Maxout Networks [10]			0.45%
Network in Network [21]			0.47%
Deeply-Supervised Networks [20]			**0.39%**

Table 4: Classification error on SVHN. Note that the student networks for all methods are identical. The results of "Teachers" are referred to the teacher network with smallest classification error.

Algorithm	# layers	# params	Error
Teachers	5	~4.9M	2.40%
Ours	19	~1.5M	**2.39%**
Student-teacher learning paradigm			
FitNets [26]			2.43%
Knowledge Distillation [15]			2.53%
State-of-the-art methods			
Maxout Networks [10]			2.47%
Network in Network [21]			2.35%
Deeply-Supervised Networks [20]			**1.92%**

than other student-teacher learning methods, and is comparable to the state-of-the-art methods. Also note the training only with standard backpropagation *w.r.t.* labels performs poorly, which in a way validates the necessity of the teacher networks' guidance.

4.2.3 SVHN. The Street View House Numbers (SVHN) dataset consists of 32×32 color images of house numbers collected by Google Street View. There are 73,257 digits in the training set, 26,032 digits in the test set and 531,131 extra training examples. Following [10, 28], we build the validation set by selecting 400 examples per class from the training set and 200 examples per class from the extra set. Then the remaining 598,388 examples in train and extra datasets are used for training. In this experiment, we only trained the model on the remaining examples and the validation dataset was used just for the parameter selection. Also we followed [10] to preprocess the dataset by Local Contrast Normalization (LCN). Similarly, we followed the maxout convolutional network reported in [10] as a teacher network, which has 3 convolutional maxout layers followed by a fully-connected layer and a softmax layer with 64-128-128-400-10 units respectively. Then we trained two additional teacher networks as CIFAR-10 did. The student network consists of 17 convolutional layers, a fully-connected layer and a softmax layer. The 11-th layer of the student network acted as the intermediate layer guided by the teacher networks. As for the determination of three teacher networks, we adopted the rule Eq.(9) in the candidate layer index set $\{2, 3\} \times \{2, 3\} \times \{2, 3\}$. From the reported results in Table 4, our student network has the comparable (or even better) performance with the top teacher network with only roughly 1/3 of parameters. Also our method is comparable to other state-of-the-art methods, and superior to some student-teacher learning methods, including FitNets and knowledge distillation.

5 CONCLUSION

This paper presents a method to train a thin deep student network, which is capable of incorporating multiple teacher networks and transferring their inner informative knowledge to help its training. To our best knowledge, this is the first attempt to combining the knowledge of multiple teacher networks in the intermediate representations. Concretely, we suggest the relative dissimilarity among examples is important and valuable for the intermediate representations learned by various teacher networks. Since multiple teacher networks usually have distinct relative dissimilarity relationships, we propose the voting strategy to unify them, which fulfills the incorporation of multiple teacher networks in intermediate layers. Therefore, with the integrated dark knowledge in the output layer, in triplet fashion the student network is guided in intermediate layers by maintaining these relative dissimilarity relationships provided by the multiple teachers. We conducted several experiments on four benchmark networks to validate the effectiveness of our proposed method. Empirical results indicate that with incorporating multiple teacher networks, the student network's performance can be significantly improved with much fewer parameters, even outperforming all the teacher networks. Besides, it also shows our method is superior to those state-of-the-art single-teacher methods, implying the advantages of combining multiple teacher networks as well.

ACKNOWLEDGMENTS

We greatly thank the anonymous referees for their valuable comments and helpful suggestions. The work is supported by the National Natural Science Foundation of China under Grant No.: 61375026 and 2015BAF15B00, and Australian Research Council Projects FT-130101457, DP-140102164 and LP-150100671.

REFERENCES

[1] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *Advances in neural information processing systems*. 2654–2662.
[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
[3] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, and others. 2007. Greedy layer-wise training of deep networks. *Advances in neural information processing systems* 19 (2007), 153.
[4] Cristian Buciluff, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 535–541.
[5] Wenlin Chen, James T Wilson, Stephen Tyree, Kilian Q Weinberger, and Yixin Chen. 2015. Compressing Neural Networks with the Hashing Trick.. In *ICML*. 2285–2294.
[6] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830* (2016).

[7] Misha Denil, Babak Shakibi, Laurent Dinh, Nando de Freitas, and others. 2013. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems*. 2148–2156.

[8] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. 2009. The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training.. In *AISTATS*, Vol. 5. 153–160.

[9] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115* (2014).

[10] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C Courville, and Yoshua Bengio. 2013. Maxout Networks. *ICML (3)* 28 (2013), 1319–1327.

[11] Çalar Gülçehre and Yoshua Bengio. 2016. Knowledge matters: Importance of prior information for optimization. *Journal of Machine Learning Research* 17, 8 (2016), 1–32.

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

[13] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*. 1135–1143.

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

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[16] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[17] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*. Springer, 84–92.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[19] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*. ACM, 473–480.

[20] Chen-Yu Lee, Saining Xie, Patrick W Gallagher, Zhengyou Zhang, and Zhuowen Tu. 2015. Deeply-Supervised Nets.. In *AISTATS*, Vol. 2. 5.

[21] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

[22] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. 2015. Sparse convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 806–814.

[23] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. 2015. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643* (2015).

[24] Patrick McClure and Nikolaus Kriegeskorte. 2016. Representational Distance Learning for Deep Neural Networks. *Frontiers in Computational Neuroscience* 10 (2016).

[25] Sharad Nandanwar and MN Murty. 2016. Structural neighborhood based classification of nodes in a network. *KDD. ACM* (2016).

[26] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* (2014).

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 815–823.

[28] Pierre Sermanet, Soumith Chintala, and Yann LeCun. 2012. Convolutional neural networks applied to house numbers digit classification. In *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 3288–3291.

[29] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.

[31] Liwei Wang, Chen-Yu Lee, Zhuowen Tu, and Svetlana Lazebnik. 2015. Training deeper convolutional networks with deep supervision. *arXiv preprint arXiv:1505.02496* (2015).

[32] Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. 2016. CNNpack: Packing Convolutional Neural Networks in the Frequency Domain. In *Advances In Neural Information Processing Systems*. 253–261.

[33] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. 2012. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*. Springer, 639–655.

[34] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* (2013).

[35] Chang Xu, Dacheng Tao, and Chao Xu. 2014. Large-margin multi-viewinformation bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (2014), 1559–1572.

[36] Chang Xu, Dacheng Tao, and Chao Xu. 2015. Multi-view intact space learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 12 (2015), 2531–2544.

[37] Wenlu Zhang, Rongjian Li, Tao Zeng, Qian Sun, Sudhir Kumar, Jieping Ye, and Shuiwang Ji. 2015. Deep Model Based Transfer and Multi-Task Learning for Biological Image Analysis. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1475–1484.