Coupled End-to-end Transfer Learning with Generalized Fisher Information

Shixing Chen[1] Caojin Zhang[2] Ming Dong[1]
[1]Department of Computer Science [2]Department of Mathematics
Wayne State University
{schen, czhang, mdong}@wayne.edu

Abstract

In transfer learning, one seeks to transfer related information from source tasks with sufficient data to help with the learning of target task with only limited data. In this paper, we propose a novel Coupled End-to-end Transfer Learning (CETL) framework, which mainly consists of two convolutional neural networks (source and target) that connect to a shared decoder. A novel loss function, the coupled loss, is used for CETL training. From a theoretical perspective, we demonstrate the rationale of the coupled loss by establishing a learning bound for CETL. Moreover, we introduce the generalized Fisher information to improve multi-task optimization in CETL. From a practical aspect, CETL provides a unified and highly flexible solution for various learning tasks such as domain adaption and knowledge distillation. Empirical result shows the superior performance of CETL on cross-domain and cross-task image classification.

1. Introduction

In computer vision, deep learning models such as Convolutional Neural Networks (CNNs) have successfully been applied to analyzing images, e.g., ImageNet [22], and achieved superior performance than other machine learning methods. However, such advances are often on account of the availability of a large amount of labeled training data. In many cases, manually labeling data can be very expensive, and when the labeled data is limited, CNN's performance will be compromised.

Transfer learning provides a framework to address this challenge. In transfer learning, one seeks to transfer related information from source tasks with sufficient data to help with the learning of target task with only limited data [29]. Recently, the ability to learn and transfer representations in CNN models has been shown to be important and effective [11, 34]. In [39], the transferability of features from various layers in neural networks was discussed. More recently, in [25], several factors (including width, depth, density, etc.) affecting the transferability for CNNs were compared.

As a special case of transfer learning, domain adaptation considers the problem when no labels of the target domain are available. It assumes that only source domain is labeled, and source and target domains have different distributions (domain discrepancy) but share the same task [29]. In recent years, various works [12, 10, 13, 8, 26] attempt to address the domain adaptation problem for deep CNNs. Usually, the domain discrepancy is modeled using Kullback-Leibler divergence or Maximum Mean Discrepancy (MMD). Then, a target domain network is fine-tuned from the source network by jointly minimizing the source domain classification error and the domain discrepancy. However, due to the relative low model accuracy and extra optimization procedures, domain adaptation remains a challenging research problem.

Knowledge distillation [17] can be considered as another special case of transfer learning, in which the knowledge from a teacher CNN is transferred to a much more concise student CNN by emulating teacher's soft-targets (a variation of softmax outputs). In this setting, teacher and student networks share the same data distribution and classification objectives. Later, FitNets [30] was proposed to include the transfer between intermediate feature maps of CNNs to improve the performance of the student CNN.

In this work, we propose a Coupled End-to-end Transfer Learning (CETL) framework to transfer the knowledge between CNNs for related tasks, and address the issues caused by domain discrepancy. Our major contributions are summarized as follows:

- CETL provides a unified transfer learning solution that can also be adapted for knowledge distillation and domain adaptation tasks, while prior works typically only consider one of these problems. In addition, through its novel architecture, CETL has great flexibility on the choice of the source network and on the architecture of the target network.
- Different from most prior work on transfer learning, the training of CETL neither uses the source data nor directly tunes on the source network. From a computation perspective, this is critical as the source dataset is usually large, and the pre-trained source network can be very big, both leading to a long training time.

2575-7075/18 $31.00 © 2018 IEEE
DOI 10.1109/CVPR.2018.00455

- We propose a novel loss function, the coupled loss, for CETL training. From a theoretical point of view, we demonstrate the rationale of the new loss function by establishing a learning bound for CETL.

- We introduce the Generalized Fisher Information (GFI) to improve multi-objective optimization in CETL. GFI conducts a dynamic allocation of shared and private weights for multi-tasks to overcome the catastrophic forgetting and preserve useful parameters for the new task. Empirical result shows the superior performance of CETL on cross-domain and cross-task image classification.

The rest of this paper is arranged as follows. In Section 2, we briefly review related work in transfer learning and its applications on image classification. In Section 3, we introduce the architecture of CETL, give the definition of GFI, and demonstrate the theoretical soundness of the coupled loss employed in CETL training. In Section 4, we present our image classification results on benchmark datasets. Finally, we conclude in Section 5.

2. Related Work

Deep CNNs achieved state-of-the-art performance in a wide range of tasks and applications in computer vision. However, in supervised learning of a CNN, a large amount of labeled data is necessary, or the model may encounter generalization issues. Thus, how to transfer useful knowledge from a source network to boost the performance of a target network with limited labeled data becomes an important research topic. In transfer learning [29, 23], we aim to learn a new task in a domain of interest called target domain when we only have sufficient data to learn a similar but different task on a source domain with different data distribution. A learning bound was introduced by [2], which claimed the error of target task is bounded by the sum of the error of the task on source and the domain discrepancy.

The research of transfer learning on deep CNN emerged recently. Yosinski et al. [39] gave one of the earliest empirical study about the feature transferability in various layers of CNN. Littwin et al. [25] proposed a framework to transfer the source data representation learned using a set of orthogonal classifiers. Azizpour et al. [1] discussed several factors influencing the transferability of features learned by CNN.

Knowledge distillation can be considered as a special case of transfer learning, in which the features learned by a teacher network are exploited to improve the performance of a relatively concise student network for the same task. Hinton et al. [17] adopted soft-targets to distill knowledge from a series of ensemble of CNNs into a single model. Following Hinton's work, Romero et al. added a difference loss between two intermediate layers to improve the performance [30]. In [38], Yim et al. defined the distilled knowledge as the Flow of Solution Procedure (FSP) matrix where the training of the student network was implemented by mimicking the FSP matrices generated by the teacher.

For domain adaptation, prior work focuses on improving deep learning models when domain discrepancy arises. A direct way is to reweigh or select samples from the source domain that are similar to the ones in the target domain [12, 9]. Rendering synthetic data is an alternative. Recently, Bousmalis et al. [3] adopted the Generative Adversarial Networks to transform source images into the target style. Most deep domain adaptation works resolve the training problem by jointly minimizing the source label classification error and the domain discrepancy. Ganin and Lempitsky [8] addressed domain discrepancy by training a CNN that minimizes the loss of label classification while maximizing the loss of a domain classifier in an end-to-end style. Weighted Maximum Mean Discrepancy (WDA) [37] was proposed later to take class weight bias into account. Tzeng et al. [35] proposed the Adversarial Discriminative Domain Adaptation (ADDA) method, where the label classifier and domain classifier are trained separately in an adversarial manner.

The proposed CETL framework is motivated by two considerations. The first one is to gradually tweak feature representations through target data reconstruction to minimize domain disparity. As in [4], Chopra et al. mitigated the domain discrepancy by layer-wise pre-training a CNN using a series of autoencoders. Later, Ghifary et al.[10] designed the model combining a traditional CNN for source label prediction with a convolutional autoencoder for target data reconstruction. The second is that the non-linear mapping between cross-modal data provides helpful deep feature representation for robust object detection with various backgrounds. For example, Xu et al. addressed the pedestrian detection problem under adverse illumination conditions, in which they exploited features in the non-linear mapping from RGB image to its corresponding thermal data [36]. Mao et al.[27] proposed a HyperLearner, which is an architecture that reconstructs various channel features (e.g., apparent-to-semantic features, temporal features and depth features) while performing pedestrian detection. In CETL, by the multi-task of simultaneous classification and reconstruction, a pre-trained source network exploits the target data for cross domain feature generation. Further, it is coupled with the target network to reconstruct those features while performing classification on the target data.

3. Coupled End-to-end Transfer Learning

In this section, we provide details on CETL. First, we show the architecture of CETL and explain the learning procedure with the coupled loss. Then, GFI is introduced for dynamic allocation of shared and private weights in multi-

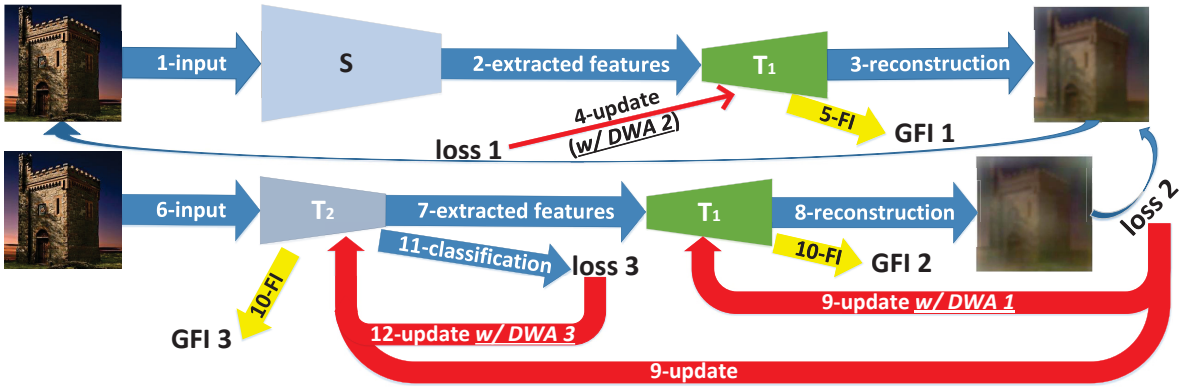

Figure 1. The CETL framework. The numbers in the figure indicate the detailed training steps of CETL.

task learning. The theoretical analysis for the coupled loss follows immediately. Last, we show how CETL can be adapted to various tasks of transfer learning, knowledge distillation and domain adaptation.

3.1. The Architecture

As shown in Fig. 1, CETL mainly consists of two CNNs with softmax outputs (source and target) that connect to a shared decoder T_1 containing deconvolution and unpooling layers for reconstruction. The pre-trained source CNN, denoted as S, aims at extracting cross domain features. The detailed steps of training in CETL are given by the numbers in the figure. Specifically, by passing the target data through S, we obtain the feature maps from each layer in S. Then, by connecting a specific layer in S to a reversed target CNN T_1, we consider S as an encoder and T_1 as a decoder. In training, we update the weights in T_1 with the reconstruction loss while keeping the weights in S unchanged. Since the feature maps in S reflect the activations of the source CNN with the input of target data, by decoding these feature back into the input space, T_1 is updated to represent the weights encoded in S in a backward manner.

Denote the datasets of target and source domains as $D_{\text{src}} = \{x_{\text{src}}, y_{\text{src}}\}$ and $D_{\text{tgt}} = \{x_{\text{tgt}}, y_{\text{tgt}}\}$ with distribution P and Q, respectively. The source CNN S is pre-trained using a supervised cross entropy loss based on the source data:

$$L_c^s(S(\theta_S), x_{\text{src}}, y_{\text{src}}) = \frac{1}{N}\sum_{i=1}^{N} \log P(y_{\text{src}}^i | x_{\text{src}}^i, \theta_S) \quad (1)$$

where θ_S denotes the model parameters while T_1 is trained using an unsupervised reconstruction loss on the target data:

$$L_r^s(T_1(\theta_{T_1}), x_{\text{tgt}}) = |T_1 \circ S(x_{\text{tgt}}) - x_{\text{tgt}}|^2 \quad (2)$$

By doing so, we find an underlying feature representation across two datasets D_{src} and D_{tgt}. It is decoded in the reconstruction $T_1 \circ S(x)$, which is a resemblance to the channel features in [27].

The target CNN denoted as T_2 is also connected with decoder T_1 to conduct the coupled learning, in which the following combined loss is minimized:

$$\lambda L_c^t(T_2(\theta_{T_2}), x_{\text{tgt}}, y_{\text{tgt}}) + (1-\lambda) L_r^t(T_1(\theta_{T_1}), T_2(\theta_{T_2}), x_{\text{tgt}}) \quad (3)$$

where

$$L_c^t(T_2(\theta_{T_2}), x_{\text{tgt}}, y_{\text{tgt}}) = \frac{1}{N}\sum_{i=1}^{N} \log P(y_{\text{tgt}}^i | x_{\text{tgt}}^i, \theta_{T_2}) \quad (4)$$

is the classification loss on the target data, and

$$L_r^t(T_1(\theta_{T_1}), T_2(\theta_{T_2}), x_{\text{tgt}}) = |T_1 \circ T_2(x_{\text{tgt}}) - T_1 \circ S(x_{\text{tgt}})|^2 \quad (5)$$

is the loss of reconstructing output of $T_1 \circ S$ using x_{tgt} as the input.

Learning: The classifier of S is pre-trained using D_{src}. First, we train the $T_1 \circ S$ using unlabeled D_{tgt}. Then, we train the target network T_2 by optimizing the combination of classification loss $L_c^t(T_2(\theta_{T_2}), x_{\text{tgt}}, y_{\text{tgt}})$ using labelled D_{tgt} and the reconstruction loss $L_r^t(T_1(\theta_{T_1}), T_2(\theta_{T_2}), x_{\text{tgt}})$ using all D_{tgt}. Alternatively, CETL can be trained in an end-to-end style. That is, we train the coupled networks using sum of all losses (coupled loss) simultaneously:

$$\begin{aligned} L_{coupled} \quad &= \lambda_1 L_r^s(T_1(\theta_{T_1}), x_{\text{tgt}}) + \lambda_2 L_c^t(T_2(\theta_{T_2}), x_{\text{tgt}}, y_{\text{tgt}}) \\ &+ \lambda_3 L_r^t(T_1(\theta_{T_1}), T_2(\theta_{T_2}), x_{\text{tgt}}) \end{aligned} \quad (6)$$

where $\lambda_i (i = 1, 2, 3)$ denotes the constant weights. We will demonstrate the rationale of the coupled loss in Section 3.3.

3.2. Generalized Fisher Information

We introduce GFI as a novel contribution for transfer learning in this section. Different from previous work [20], we take the correlation of two tasks into account and dynamically allocate shared and private weights for the corresponding tasks.

In CETL, the coupled networks contain multi-objectives with shared parameters. That is,

$$L_r^s(T_1(\theta_{T_1}), x_{\text{tgt}}) \quad \text{and} \quad L_r^t(T_1(\theta_{T_1}), T_2(\theta_{T_2}), x_{\text{tgt}}) \quad (7)$$

share parameters in T_1.

$$L_c^t(T_2(\theta_{T_2}), x_{\text{tgt}}, y_{\text{tgt}}) \quad \text{and} \quad L_r^t(T_1(\theta_{T_1}), T_2(\theta_{T_2}), x_{\text{tgt}}) \quad (8)$$

share parameters in T_2.

Thus, the issues of catastrophic forgetting tend to override model parameters learned in previous tasks, leading to impaired performance [20].

Fisher information (FI) is a way of measuring the amount of information that an observable random variable X carries for an unknown parameter θ of a distribution that models X. In [20], Fisher information F_i with respect to certain weights θ_i of a neural network is derived from the cross-entropy loss and used to measure parameters' importance to a given task. The Elastic Weight Consolidation loss using FI from the given task is designed as a regularization to keep the weights with large F_i unchanged in order to avoid catastrophic forgetting.

However, when all parameters are determined as important by the prior task, update of weights with respect to the new task will be trivial. The training of the new task may fail to converge. Thus, we introduce a new measure, Relative Fisher (RF) information, to determine the correlation of FIs derived from the two tasks. Denote the losses for two tasks as L_1 and L_2, with shared parameters θ_i, $i = 1, \cdots, m$, where m denotes the total number of parameters, we have:

$$RF_i = I(F_{1,i}, F_{2,i} | \theta_i^*) \quad (9)$$

where $I(\cdot,\cdot)$ denotes the mutual information normalized in $[0,1]$, $F_{1,i}$ and $F_{2,i}$ are random variables representing the FI with respect to L_1 and L_2, respectively. The higher RF_i is, the more probable the two tasks may share the weights θ_i. Finally, the Generalized Fisher Information (GFI) is defined as:

$$GFI_i = \begin{cases} \begin{cases} 0 & \text{with probability } p \\ F_i & \text{with probability } 1-p \end{cases} & \text{if } RF_i < u \\ \begin{cases} 0 & \text{with probability } 1-p \\ F_i & \text{with probability } p \end{cases} & \text{if } RF_i \geq u \end{cases} \quad (10)$$

where RF_i is used to indicate whether two tasks should share the same weights, and the hyperparameters u and p are set at 0.5 and 0.9, respectively. Specifically, if $RF_i \geq 0.5$, weights will be shared, and we set the $GFI_i = 0$ with a low probability 0.1 to retain flexibility. Otherwise ($RF_i < 0.5$), F_i has a high probability 0.9 to be dropped, and thus the new task can be better learned without regularization.

We define the Dynamic Weight Allocation (DWA) loss as the regularization term:

$$DWA = \sum_i \frac{\lambda}{2} GFI_i (\theta_i - \theta_i^*)^2 \quad (11)$$

which allows dynamic allocations of shared and private parameters for different tasks. We apply it on the joint optimization of the multi-objectives in Eqs. (7) and (8).

3.3. Theoretical Analysis of the Coupled Loss

In this section, we derive an error bound for CETL learning, which provides a rigorous theoretical explanation on the rationale for the coupled loss function adopted in CETL (Eq. (6) in Section 3.1).

We assume the ground truth concept for the source and target as c_{src} and c_{tgt}, respectively. Denote $T_1 \circ T_2(x)$ for $x \in P$ as $T_1 \circ T_2 \in P$, and for $x \in Q$ as $T_1 \circ T_2 \in Q$, respectively. The similar notations work for $T_1 \circ S$ as well. We denote all constant numbers in proofs as C for simplicity. In addition, E_Z denotes the expectation on distribution Z, and $\sup_f$ represents taking the maximum value over the collection of functions f.

Lemma 1 *If $E_P|S - c_{src}| \leq C$, $C > 0$, $E_Q|c_{src} - c_{tgt}| \leq \lambda_1$, $\sup_f |E_P f - E_{T_1 \circ S \in P} f| \leq \lambda_2$, for any $f \in P, Q$, $\lambda_1, \lambda_2 > 0$, then there exists some constant $C > 0$, for any measurable function $f \in P, Q$, $f > 0$,*

$$\begin{aligned} E_Q|T_2 - c_{tgt}| \quad &\leq C + |E_{T_1 \circ S \in P} f - E_Q f| + E_Q|T_2 - S| \\ &+ 2\sup_f |E_{T_1 \circ S(x) \in Q} f - E_{x \in Q} f| \\ &+ 2\sup_f |E_{T_1 \circ T_2(x) \in Q} f - E_{x \in Q} f| \end{aligned} \quad (12)$$

Proof: See Supplementary Material A.1.

Lemma 2 *Assume $E_{T_1 \circ S \in P}|c_{src} - c_{tgt}| \leq C$, $E_{T_1 \circ S \in Q}|c_{src} - c_{tgt}| \leq C$, $E_{T_1 \circ S \in P}|c_{tgt}| \leq C$, $E_{T_1 \circ S \in Q}|c_{src}| < C$, for some $C > 0$, then there exists some constant $C > 0$, such that for any measurable function $f > 0$, and $f \in P, Q$,*

$$\begin{aligned} \sup_f |E_{T_1 \circ S \in P} f - E_Q f| \quad &\leq C + E_{T_1 \circ S \in P}|S - c_{src}| \\ &+ \sup_f |E_{T_1 \circ S \in Q} f - E_Q f| \\ &+ E_Q|T_2 - S| \end{aligned} \quad (13)$$

Proof: See Supplementary Material A.2.

Theorem 3 *If all conditions in Lemma 1 and 2 hold. We have the bound for CETL as:*

$$\begin{aligned} E_Q|T_2 - c_{tgt}| \quad &\leq E_P|S - c_{src}| + 2E_Q|T_2 - S| \\ &+ 3\sup_f |E_{T_1 \circ S(x) \in Q} f - E_{x \in Q} f| \\ &+ 2\sup_f |E_{T_1 \circ T_2(x) \in Q} f - E_{x \in Q} f| \end{aligned} \quad (14)$$

Proof

Combing the results of Lemma 1 and 2, the desired result follows.

Remark 4 The left hand side (LHS) of Eq. (14) is the expected classification error on the target domain. It is the ultimate objective to be minimized, but direct optimization is virtually impossible. **This is our motivation and rationality to provide our theoretical analysis for the error bound**. Specifically, we derive the upper bound of LHS as the right hand side (RHS) in Eq. (14) and proposed to minimize RHS instead. More importantly, it guides us to define

the coupled loss in Eq. (6). Specifically, RHS in Eq. (14) and Eq. (6) correspond as follows. Classification loss: the first term in RHS and the second term in Eq. (6); Cross domain loss: the second term in RHS and the third term in Eq. (6); And reconstruction loss: the last two terms in RHS and the first term in Eq. (6).

3.4. Algorithms of CETL

In this section, we will show that CETL is a unified framework that can be adapted into different tasks of transfer learning, knowledge distillation and domain adaptation. Moreover, CETL outperforms these instantiations by incorporating GFI. For better illustration, we first give the pseudo code of CETL for transfer learning in Algorithm 1, and then we show its variants.

Algorithm 1 Algorithm of CETL

procedure STAGE 1
  top:
  Input x_{tgt}
  feature encoding $\leftarrow S$
  feature decoding $\leftarrow T_1$
    loss1, recons1 $\leftarrow$ *reconstruct* x_{tgt}
    update $T_1 \leftarrow loss1(w/DWA2)$
    GFI1 $\leftarrow$ *recons1*
procedure STAGE 2
  Input x_{tgt}
  feature encoding $\leftarrow T_2$
  feature decoding $\leftarrow T_1$
    loss2, recons2 $\leftarrow$ *reconstruct recons1*
    update $T_1 \circ T_2 \leftarrow$ *loss2 w/ DWA1*
    GFI2, GFI3 $\leftarrow$ *recons2*
procedure STAGE 3
  Input x_{tgt}
  classification loss3 $\leftarrow T_2$
    update $T_2 \leftarrow$ *loss3 w/ DWA3*
  if *loss1, loss2, loss3* not converged
    goto *top*
  end if

The rationale for the three-stage training in Algorithm 1 is given below. In the coupled loss (Eq. 6), there are three loss terms. According to [20], catastrophic forgetting happens in multi-task training. If we optimize $L_{coupled}$ directly using SGD, weights learned by certain tasks can be overridden by others, leading to the failure of convergence on these tasks. Thus, we carefully designed an iterative three-stage training, in which GFI is introduced to indicate the importance of weights learned in the previous task, and DWA loss is applied as a regularization to remember the important ones during updating.

The main drawback of FI in [20] is that if most of the weights are considered important by the previous task, the model's ability to learn a new task will be dramatically weakened. Differently, GFI uses hyperparameters to determine if the new task learning should be affected by FI. We define DWA loss using GFI to allow a dynamic allocation of shared and private weights for all the tasks.

3.4.1 Transfer Learning

When we consider a traditional transfer learning problem, T_2 has an architecture similar to S, and T_1 has the one with reversed layers. As shown in Algorithm 1, we have three learning stages in total. For the first stage, with S pre-trained on the source data, we reconstruct the target data with $T_1 \circ S$, in which the weights in S are frozen while T_1's are updated by the reconstruction loss to simulate S in the reversed order. After the reconstruction by $T_1 \circ S$, we can obtain $GFI1$, the GFI for the weights in T_1 with respect to the reconstructed output $recons1$.

During the second stage, we transfer the information in T_1 to T_2 while incorporating $DWA1$. Specifically, by passing the target data through $T_1 \circ T_2$, we get the reconstruction $loss2$. We use $loss2$ to update T_2 and $loss2$ with $DWA1$ to update T_1. In this way, we keep weights unchanged if they were considered important by $GFI1$ in $T_1 \circ S$, and update the other weights for the reconstruction in $T_1 \circ T_2$. At the end of this stage, we can obtain $GFI2$ and $GFI3$, which quantify the gradients of outputs with respect to weights in T_1 and T_2, respectively. Later, $DWA2$ will be incorporated with $loss1$ to update $T_1 \circ S$.

In the third stage, we have the classification loss on the target data given by T_2, and we update T_2 with this loss using $DWA3$ as the regularization. Thus, part of the weights in T_2 will be updated for reconstruction while the rest would be for classification. The three stages are repeated iteratively until all losses are converged.

3.4.2 Knowledge Distillation

In knowledge distillation [17], teacher (source) and student (target) networks are generally assumed to share the same dataset. To adapt CETL for knowledge distillation, we simply need to let T_1 be a much more concise architecture comparing with S and let T_2 have the reversed layers of T_1. Furthermore, with CETL, we can also handle the situation when source and target have different datasets. Actually, we don't need the source data (usually a large dataset) for (expensive) training as long as we can utilize the weights from S to take advantage of the soft targets.

As an improvement, FitNets was proposed later to utilize not only the soft targets but also the feature maps from the middle layer of the S network [30]. CETL can be similarly modified for FitNets and we will not repeat it here due to space limit.

3.4.3 Domain Adaptation

The major issue we need to resolve when using CETL for domain adaptation is regarding the amount of labeled data in the target domain. We consider the following two scenarios: 1) When we have limited training labels for the target domain, we can still use them to compute the classification loss in the third stage of learning. As for the reconstruction losses, we can incorporate the reconstruction of testing samples in the target domain, similar to other domain adaptation methods [10], to improve the performance. 2) For the extreme case when no training labels are available, based on prior work in [10], we will have to use the training data from source domain to update the networks with the classification loss. In this way, the features in T_2 are considered invariant for both source and target domains, and thus the classification performance on target domain can be improved. Specifically, Algorithm 1 will be modified as follows: we will use source data x_{src} as the input in line 17 instead of x_{tgt}.

3.4.4 Advantages of CETL

The advantages of CETL over existing transfer learning models can be summarized as follows:

- Comparing with directly fine-tuning on S, CETL can handle the situation when target data are not sufficient to update a deep/big source network.
- By incorporating GFI, CETL keeps the useful weights for reconstruction while updating the others. This leads to higher efficiency and better performance.
- From a practical perspective, CETL provides a very high level of flexibility on the selection of source networks. Regardless of the source architecture, source data availability, and the choice of computing platform, CETL can always leverage the pre-train source network for performance gain as long as the source output can be obtained with a forward pass. No retraining or fine-tuning is required. This unique nature makes CETL highly practical in solving various real-world problems.

We show these advantages through extensive experiments in the next section.

4. Experiments

In this section, we conduct experiments from three aspects to show the superior performance and flexibility of CETL. First, for general transfer learning tasks, we demonstrate the functionality of the components in the CETL algorithm and validate the configuration of CETL, followed by the performance analysis of the preferred architecture on various scenarios. Then, we compare CETL with other models on the performance of knowledge distillation task, and explicitly explain the rationale of using GFI. Last, we compare CETL with other state-of-the-art models on domain adaptation experiments.

In the experiments, we adopt widely used benchmark datasets to evaluate the performance of CETL, including CIFAR-10 (CI) and CIFAR-100 [21], STL-10 (ST) [5], ImageNet [6], MNIST (MN) [24], USPS (US) [18] and SVHN (SV) [28]. The descriptions of these datasets are given in Table 1, and we explain how to use them in different tasks in the following sections.

4.1. Transfer Learning

As mentioned before, both knowledge distillation and domain adaptation can be considered as special cases of transfer learning. To avoid any confusion, in this section, we consider the scenario where both domains and tasks are different between the source and the target.

4.1.1 Configurations of CETL

To start with, we consider the transfer between ImageNet and CIFAR-10 to decide the preferred configuration of CETL in Theano [33]. For ImageNet, we use the trained AlexNet model [22] provided by Caffe [19] as S. For CIFAR-10, as generally handled in transfer learning approaches, we randomly select only 20% of the original training data while keeping all the original testing data to form a subset of CIFAR-10 called CIFAR-10-s as the target dataset. Also, since the input image size in CIFAR-10 is much smaller than that in ImageNet, we adopt a reduced AlexNet and call it CI-CNN.

In CI-CNN, there are still five convolutional layers and three fully-connected layers, but the numbers of kernels in each layer are all reduced to about 1/2 to 1/4 of the ones in AlexNet. Also, the convolutional kernels are all set to be 3×3. As for the reverse architecture of CI-CNN, there are three fully connected layers followed by alternative unpooling and deconvolution layers. For each reverse layer, the number of kernels is the same as the one in the corresponding layer in CI-CNN.

Specifically, we resize and pass the training data in CIFAR-10-s to the trained AlexNet to extract features before the last fully connected layer. Then, T_1 with the reverse architecture of CI-CNN reconstructs the feature from S. After that, the CI-CNN in T_2 carries out the multi-objective optimization to simultaneously reconstruct the CIFAR-10-s images with T_1 and classify them into ten image categories.

Table 2. Comparison of different configurations of CETL.

Baseline	CETL$_f$	CETL$_u$	CETL$_e$	CETL$_{fi}$	CETL
61.29%	62.41%	62.63%	63.97%	64.19%	**65.33%**

Table 1. The properties of the benchmark datasets adopted in the experiments.

	CIFAR-10	STL-10	CIFAR-100	ImageNet	MNIST	USPS	SVHN
# of classes	10	10	100	1000	10	10	10
Purpose	image classification		image classification		digit recognition		
Training samples	50000	5000	50000	1.2 million	60000	7291	73257
Testing samples	10000	8000	10000	100000	10000	2007	26032
Image type	color	color	color	color	grayscale	grayscale	grayscale
Image size	32×32	96×96	32×32	256×256 (resized)	28×28	16×16	32×32

In Table 2, we compared different settings of CETL for the classification accuracy of testing samples in CIFAR-10-s. The baseline accuracy shows the result of directly training on CI-CNN. For CETL$_f$, we train $T_1 \circ S$ until convergence and then train $T_1 \circ T_2$ with T_1 fixed. In this case, the reconstruction objective of T_2 can only be partially fulfilled since half of the weights in $T_1 \circ T_2$ are not updated. CETL$_u$ takes a step further to update $T_1 \circ T_2$ after $T_1 \circ S$ converged, but the problem is that T_1 could be tuned as a convolutional auto-decoder without maintaining the knowledge learned from S. In CETL$_e$, we update $T_1 \circ S$ and $T_1 \circ T_2$ iteratively until T_1 converges for both reconstruction objectives. However, without the control of GFI, all the weights in T_1 and T_2 are updated in the same way regardless of their importance for a given task.

It is also clear from Table 2 that the performance of CETL$_e$ can be improved with FI, but the gain of CETL$_{fi}$ is not much. Finally, CETL with GFI dynamically allocates the weights in T_1 and T_2 to either shared or private, and updates them according to their importance for various tasks. Obviously, the best performance is achieved by CETL with GFI.

Notice that in this experiment, we neither update S which is more complicated than T_1, nor use the source dataset, ImageNet, which is dramatically larger than CIFAR-10-s. Instead, we take advantage of the trained AlexNet to improve the performance on CIFAR-10-s. In the following, we denote the selected configuration, CETL with GFI, as CETL, and use it in all the experiments. The architectures of S, T_1 and T_2 will be modified for different tasks.

4.1.2 Different Source Networks

To show the flexibility of CETL, we perform the experiments with various combinations of source and target networks. In Table 3, all source architectures except for CI-CNN are pre-trained networks for the classification of ImageNet [19], and then transferred to CIFAR-10-s and STL-10 respectively with CETL to improve their performance. CI-CNN was trained on CIFAR-100 from scratch and used as one of the source networks for STL-10. We used the same architecture CI-CNN for both CIFAR-10-s and STL-10 as the target network. As a comparison, we obtained the baseline accuracy by directly training on CI-CNN. In addition, we replaced the last fully-connected layer in VGG and fine-tuned it using the target datasets. The accuracy is reported as FT-VGG.

Table 3. Comparison of different source networks.

	CIFAR-10-s+ CI-CNN	STL-10+ CI-CNN
AlexNet [22]	65.33%	62.98%
VGG [31]	**65.57%**	62.61%
GoogleNet [32]	65.14%	62.30%
ResNet-50 [16]	64.37%	62.77%
CI-CNN	-	**65.49%**
Baseline	61.29%	60.52%
FT-VGG	61.35%	61.17%

Apparently, fine-tuning is not as effective as CETL, providing little improvement. For CIFAR-10-s, highest accuracy is achieved when transferred from VGG. For STL-10, transferring from CI-CNN performs the best. The reason is that STL-10 dataset is more similar to CIFAR than to ImageNet. Also, it is clear that source data is a more important factor for the performance gain than the source architecture.

4.2. Knowledge Distillation

Knowledge distillation considers the problem when source and target data are the same while the student network is much smaller (thinner) than the teacher network. In this section, we compare CETL with other state-of-the-art knowledge distillation models on CIFAR-10 and CIFAR-100 datasets. To make a fair comparison, we follow some recent work [38] and choose ResNet-26 as the teacher network and CI-CNN as the student network with less than 10% parameters of AlexNet. Specifically, for CIFAR-10, the teacher architectures are exactly the same. The student networks differ, but the initial accuracy (before distillation) are very close (87.91% in [38] and 87.55% in CETL). Same holds for CIFAR-100.

As shown in Table 4, CETL outperforms other knowledge distillation models on both CIFAR-10 and CIFAR-100 datasets. As more training samples are available for each category in CIFAR-10, the improvement is marginal through knowledge distillation. However, classification accuracy is significantly increased in the case of CIFAR-100, close to the teacher performance. This mainly attributes to

Table 5. Comparison on domain adaptation. A dash means that the result is not reported by the model.

	MN-US	US-MN	SV-MN	MN-SV	ST-CI	CI-ST
Source	85.55%	65.77%	62.33%	25.95%	54.17%	63.61%
SA [7]	85.89%	51.54%	63.17%	28.52%	54.04%	62.88%
ReverseGrad [8]	91.11%	74.01%	73.91%	35.67%	56.91%	66.12%
DRCN [10]	91.80%	88.67%	81.97%	40.05%	58.86%	66.37%
ADDA [35]	89.40%	90.10%	76.00%	-	-	-
WDA [37]	72.30%	65.50%	67.30%	23.5%	-	-
CETL	**92.96%**	**90.89%**	**83.33%**	**45.27%**	**60.11%**	**66.39%**

Table 4. Comparison on knowledge distillation.

	CIFAR-10	CIFAR-100
Teacher	91.86%	65.23%
Student	87.55%	60.71%
FitNets [30]	88.57%	61.28%
Soft-targets [17]	88.45%	61.03%
FSP DNN [38]	88.70%	63.33%
CETL	**89.11%**	**64.83%**

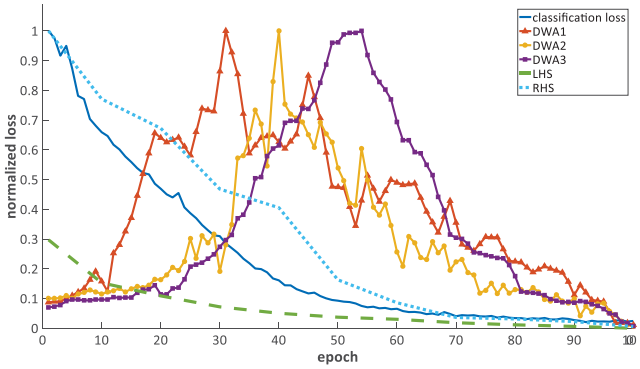

Figure 2. Learning curves of CETL.

the low number of samples per category in CIFAR-100.

To further demonstrate the rationale of the DWA loss, we trace the changes of classification loss of T_2 and normalized DWA losses by the solid lines in Fig. 2 for the CIFAR-10 classification task. Clearly, the classification loss decreases with the epochs as usual. However, note that the DWA1 loss first increases to a peak value before decreasing and getting converged. This is because at the very beginning of training, T_1 is randomly initialized and for the first a few epochs, most of the weights in T_1 are not important for $T_1 \circ S$ and thus the loss is small. Around epoch 32, the weights in T_1 becomes more important for $T_1 \circ S$, leading to a larger DWA1 loss, after which the DWA1 loss decreases as the changes of weights decrease until converged. The trends of DWA2 and DWA3 losses follow a similar pattern.

4.3. Domain Adaptation

For the last task, we compare CETL with current state-of-the-arts on domain adaptation where target data does not have labels but has same categories as the source data. In this case, similar to other models, the multi-objective in CETL is to use target data for reconstruction and source data for classification. Following the same settings used in some recent work [10], we directly compare CETL with the reported performance in the literature in Table 5. ADDA and WDA results are directly obtained from [35] and [37], and a dash in the table means that the result is not reported by the corresponding model on the given dataset.

Clearly, CETL significantly improved from the prior arts and achieved the best performance on all domain adaptation combinations. In particular, CETL with coupled loss and GFI can overcome the catastrophic forgetting in multi-tasks and outperforms models (e.g., DRCN) that consider the tasks (i.e., reconstruction and classification) separately. Furthermore, we compared CETL with associative domain adaptation models in [15, 14]. Results show that CETL is very competitive with these models on domain adaptation while having the flexibility of also performing knowledge distillation and transfer learning. For example, in "SV-MN" (one of the best results mentioned in [14]), the relative improvement is 40.86% (before and after domain adaptation) in [14], 22.16% in [15], while CETL gains 33.69%.

Finally, to validate Remark 4 from an experimental perspective, we demonstrate the normalized values of RHS and LHS in Eq. (14) w.r.t the training epochs in "SV-MN". As shown by the dashed and dotted lines in Fig. 2, LHS is bounded by RHS, and clearly, their difference converges to almost zero as the training epoch increases.

5. Conclusion

In this paper, we proposed a novel CETL framework for image classification. A novel loss function, the coupled loss, established base on the learning bound of CETL, was introduced for CETL training. In addition, GFI was integrated to improve the multi-task optimization in CETL. Experimentally, we extensively compared CETL with other state-of-the-art models for various tasks on benchmark datasets and achieved superior performance.

Acknowledgment This work was partially supported by US National Science Foundation (NSF) under grant CNS-1637312.

References

[1] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1790–1802, 2016.

[2] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[3] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[4] S. Chopra, S. Balakrishnan, and R. Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML workshop on challenges in representation learning*, volume 2, 2013.

[5] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[7] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[8] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[9] W. Ge and Y. Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[12] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 222–230, 2013.

[13] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

[14] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[15] P. Haeusser, A. Mordvintsev, and D. Cremers. Learning by association – a versatile semi-supervised training method for neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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

[18] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[20] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, page 201611835, 2017.

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Masters thesis, Department of Computer Science, University of Toronto*, 2009.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[23] I. Kuzborskij, F. Orabona, and B. Caputo. From n to n+1: Multiclass transfer incremental learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] E. Littwin and L. Wolf. The multiverse loss for robust transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[27] J. Mao, T. Xiao, Y. Jiang, and Z. Cao. What can help pedestrian detection? In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[28] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[29] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[30] 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.

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

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[33] T. T. D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[34] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3081–3088. IEEE, 2010.

[35] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] D. Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[37] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[38] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[39] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.