Transductive Semi-Supervised Deep Learning using Min-Max Features

Weiwei Shi[1], Yihong Gong[1]*, Chris Ding[2], Zhiheng Ma[1]
Xiaoyu Tao[1], and Nanning Zheng[1]

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2] University of Texas at Arlington
shiweiwei.math@stu.xjtu.edu.cn, ygong@mail.xjtu.edu.cn, chqding@uta.edu
{mazhiheng, txy666793}@stu.xjtu.edu.cn, nnzheng@mail.xjtu.edu.cn

Abstract. In this paper, we propose Transductive Semi-Supervised Deep Learning (TSSDL) method that is effective for training Deep Convolutional Neural Network (DCNN) models. The method applies transductive learning principle to DCNN training, introduces confidence levels on unlabeled image samples to overcome unreliable label estimates on outliers and uncertain samples, and develops the Min-Max Feature (MMF) regularization that encourages DCNN to learn feature descriptors with better between-class separability and within-class compactness. TSSDL method is independent of any DCNN architectures and complementary to the latest Semi-Supervised Learning (SSL) methods. Comprehensive experiments on the benchmark datasets CIFAR10 and SVHN have shown that the DCNN model trained by the proposed TSSDL method can produce image classification accuracies compatible to the state-of-the-art SSL methods, and that combining TSSDL with the Mean Teacher method can produce the best classification accuracies on the two benchmark datasets.

Keywords: Transductive Semi-Supervised Deep Learning (TSSDL), Min-Max Feature (MMF) regularization, Deep Convolutional Neural Network (DCNN), confidence levels.

1 Introduction

To date, Deep Convolutional Neural Networks (DCNNs) have shown state-of-the-art performances in numerous computer vision applications, such as image classification [1–5], object detection [6, 7], face recognition [8–10], image retrieval [11–14], etc. One of the main driving forces of these great accomplishments is the availability of large scale image datasets that contain millions of labeled training samples. However, creating a large scale, high quality training set by human labeling is very time-consuming, expensive, or even prohibitive (e.g. training set for image semantic segmentation). On the other hand, there are an unlimited number of unlabeled images on the Internet, which can be

* Corresponding author

easily obtained by web crawlers and search engines. In recent years, there have been increased research efforts that employ Semi-Supervised Learning (SSL) approaches to train DCNNs with both labeled and unlabeled image samples. Such research efforts have a great potential to dramatically reduce the cost of training DCNN models with high performance accuracies.

Many traditional SSL methods are based on the so-called label propagation approach [15–18], which measures similarities between training samples, and propagates labels of labeled samples to nearby unlabeled ones. Another line of research works are known as Transductive Semi-Supervised Learning (TSSL) [19–23], in which labels of unlabeled samples are treated as variables, and are determined through the iterative training process. At the end of the training process, a classifier is learned from both the labeled and unlabeled training samples. As additional unlabeled samples are used for training, classifiers generated by SSL and TSSL methods usually outperform their counterparts generated by supervised learning methods given the same amount of labeled training samples.

There are two common problems associated with the traditional SSL and TSSL methods. First, these methods generally require high-quality feature descriptors to measure the similarity distances among the training samples from the very beginning of the training process. This requirement makes them difficult to be applied to DCNN training, because feature descriptors generated by a DCNN model are of low quality at early training stages, and improve gradually along the iterative training process. Second, traditional SSL and TSSL methods treat every unlabeled sample equally, which makes the model learning process vulnerable to outliers and uncertain data samples. This problem will become more severe for training DCNNs, because initial feature descriptors generated by DCNNs are of low quality and unstable, which may mislead the training process into a wrong direction.

Recent research works [24–28] have explored supervisory information from unlabeled image samples by adding different perturbations to each image, and enforcing the label consistency between different perturbed versions of the image. The temporal ensembling (TempEns) work [25] enhances the perturbation-based methods by maintaining an Exponential Moving Average (EMA) of label predictions on each training sample, and penalizing the prediction of the network-in-training which is inconsistent with the corresponding EMA prediction. The Mean Teacher method [26] further improves TempEns by using EMA on DCNN model weights instead of label predictions. These two latest methods have achieved state-of-the-art image classification accuracies in the SSL field.

In this paper, we propose a novel Transductive Semi-Supervised Deep Learning (TSSDL) method that is effective for training DCNN models. The proposed TSSDL method is comprised of three major components. First, we extend the traditional TSSL methods to make it applicable to DCNN training. We also treat the labels of unlabeled samples as variables, and try to determine their optimal labels together with the optimal DCNN parameter set by minimizing the proposed loss function through the iterative training process. To the best of our knowledge, this is the first attempt in the literature to apply the trans-

ductive learning principle to DCNN model training. Second, to overcome the problem that low-quality feature descriptors generated by the DCNN model at early training stages may mislead the training process into a wrong direction, we introduce the confidence level r_i for each unlabeled sample $\mathbf{X}_i$, which indicates how reliable is the label vector $\mathbf{y}_i$ of $\mathbf{X}_i$ predicted by the current version of DCNN model. r_i is computed based on the assumption that $\mathbf{y}_i$ will be more reliable if $\mathbf{X}_i$ is located in densely populated regions, and vice versa. This is because label predictions for unlabeled samples in densely populated regions tend to be more accurate than those in sparsely populated ones. Third, we develop the Min-Max Feature (MMF) regularization that enforces features learned by the DCNN model to have the following properties: If two images possess the same label, then the distance between their feature descriptors must be minimized; otherwise, the distance must be larger than a predefined margin. The MMF regularization can be considered as an important extension to the traditional label propagation methods which mandates not only that images with the same label be close to each other in the feature space, but also that images with different labels be separated from each other by a predefined margin. These two mandates serve to force the DCNN model to learn better feature descriptors from the given labeled and unlabeled training samples.

The proposed TSSDL method is independent of any DCNN architectures, and is complementary to the latest SSL methods. Comprehensive experimental evaluations on the benchmark datasets CIFAR10 and SVHN have shown that the DCNN model trained by the proposed TSSDL method can produce image classification accuracies that are compatible to the state-of-the-art SSL methods, and that combining TSSDL with the Mean Teacher method can produce the best classification accuracies on the two benchmark datasets.

To sum up, our main contributions include:

- We extend the traditional TSSL methods to make it applicable to DCNN model training.
- We introduce the confidence level for each unlabeled sample to discount influences from outliers and uncertain samples.
- We develop the MMF regularization to make the DCNN model learn feature descriptors such that images with the same label will be close to each other in the feature space, and that images with different labels will be separated by a predefined margin.

The remaining of this paper is organized as follows: Section 2 reviews related works. Section 3 describes our method. Section 4 presents the experimental evaluations and analysis, and Section 5 concludes our work.

2 Related Work

In the past, many semi-supervised learning (SSL) methods [19, 20, 29–32] have been proposed in the literature. A large group of traditional SSL methods are

based on the label propagation approach, which infers labels for unlabeled samples by measuring similarities between training samples, and propagating labels of labeled samples to nearby unlabeled ones.

Another line of research works are known as Trasductive Semi-Supervised Learning (TSSL) [19–22]. The key characteristic of TSSL is that labels of unlabeled samples are viewed as optimization variables, and are iteratively updated in the training process. As the learning proceeds, predicted labels of unlabeled samples become more consistent among themselves, and with labels of labeled samples.

Traditional SSL and the TSSL methods assume that feature descriptors of training samples are known and fixed, and their performance accuracies are highly dependent on the quality of the provided feature descriptors. This requirement makes them difficult to be applied to DCNN model training, because in deep learning, feature descriptors are learned during the training process. They are of low quality at early stages, and are gradually improved along the training process.

In recent years, there have been increased research efforts to develop SSL methods for DCNN model training. Some works use unlabeled data to pre-train DCNN models in an unsupervised way, and then fine-tune the models on labeled data [33–36]. Other works use unlabeled data in the entire training process instead of just pre-training. For example, Hoffer *et al.* [37] used the regularization term of entropy minimization to enforce that one sample is assigned to one class to reduce the overlaps between different classes. Weston *et al.* [38] proposed an unsupervised embedding for DCNN training. Kingma *et al.* [39] proposed the deep generative models for SSL. Based on the expectation-maximization algorithm, Papandreou *et al.* [39] developed a DCNN training method for SSL semantic image segmentation. Abbasnejad *et al.* [40] proposed the infinite variational autoencoder for SSL. Haeusser *et al.* [41] proposed a SSL method by association.

There also exist research studies [42–45] that use Generative Adversarial Networks (GAN) to generate additional training samples by optimizing an adversarial game between the discriminator and the generator. Samples generated by GAN can be viewed as a kind of "data augmentation".

A line of works more closely related to our method are the regularizations of features learned by DCNNs. For example, Sajjadi *et al.* [46] proposed to use perturbations (such as random data augmentation, dropout) on images to learn robust features. Miyato *et al.* [27] proposed a virtual adversarial training (VAT) method with the virtual adversarial loss, which improves the robustness of the model's predictions against adversarial perturbations. Rasmus *et al.* [47] proposed a SSL method with ladder networks [48]. Π model [25] evaluates the network twice for each training sample under two different i.i.d perturbations at every iteration of the training process, and enforces the label predictions on the two perturbed versions of the training sample to be consistent. Temporal ensembling (TempEns) [25] enhances the Π model by maintaining an exponential moving average (EMA) of label predictions on each training sample, and using it

as the target prediction for the sample. It penalizes the prediction of the network-in-training which is inconsistent with its corresponding target prediction. Mean Teacher [26] further improves TempEns by using EMA on model weights instead of label predictions. To date, TempEns and Mean Teacher have achieved the state-of-the-art image classification accuracies in the SSL field.

However, the latest SSL methods described above only consider the perturbations around each single data point, while ignore the relationships between data points. In other words, these methods have not fully utilized the information, such as the structural information in the unlabeled data. It is known that data points belonging to the same class tend to form clusters. This has motivated us to develop the MMF regularization to utilize the structural information among unlabeled data points.

3 Methodology

3.1 Preliminaries

Let $\mathcal{D} = \mathcal{L} \bigcup \mathcal{U}$ be the entire training set, where $\mathcal{L} = \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^{L}$, $\mathcal{U} = \{\mathbf{X}_i\}_{i=L+1}^{L+U}$ denote the labeled and unlabeled sample sets, respectively, and $\mathbf{X}_i$ is the i^{th} training sample. If $\mathbf{X}_i \in \mathcal{L}$, then $\mathbf{y}_i = [y_i^1, y_i^2, \cdots, y_i^K]^\top \in \{0,1\}^K$ is the corresponding one-hot ground-truth label vector, where $y_i^j = 1$ if $\mathbf{X}_i$ belongs to the j^{th} class, and $y_i^j = 0$ otherwise. K refers to the number of classes, L, U are the numbers of labeled and unlabeled training samples, respectively. Usually $L \ll U$. Let $N = L + U$ be the total number of training samples.

3.2 Transductive Semi-Supervised Deep Learning (TSSDL)

When training a DCNN model using a Supervised Learning (SL) method, the typical loss function can be written as:

$$\ell^{\mathrm{SL}}(\mathcal{X}, \mathcal{Y}; \theta) = \sum_{i=1}^{L} \ell_0(\mathbf{X}_i, \mathbf{y}_i; \theta)\,, \tag{1}$$

where $\mathcal{X} = \{\mathbf{X}_i\}_{i=1}^{L}$, $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^{L}$, θ is the entire parameter set of the DCNN model, and $\ell_0(\mathbf{X}_i, \mathbf{y}_i; \theta)$ is the loss for sample $\mathbf{X}_i$. Here, $\mathcal{Y}$ is the manually provided ground-truth label vector set for the training set $\mathcal{X}$, and is fixed throughout the entire training process. If the softmax loss is used, which is a popular choice for most image classification tasks, then Eq. (1) can be rewritten as:

$$\ell^{\mathrm{SL}}(\mathcal{X}, \mathcal{Y}; \theta) = \sum_{i=1}^{L} \mathrm{CEsoftmax}(\mathbf{W}\mathbf{f}(\mathbf{X}_i; \theta), \mathbf{y}_i), \tag{2}$$

where $\mathbf{f}(\mathbf{X}_i; \theta)$ is the output of the DCNN's penultimate layer for sample $\mathbf{X}_i$, which can be considered as the learned feature descriptors of $\mathbf{X}_i$, and $\mathbf{W}$ is the parameters of the last fully-connected layer of the DCNN. Here $\mathrm{CEsoftmax}(\mathbf{a}, \mathbf{b}) =$

Cross-Entropy(softmax($\mathbf{a}$), $\mathbf{b}$). The goal is to learn an optimal parameter set θ^* that minimizes the loss function: $\theta^* = \arg\min_\theta \ell^{\mathrm{SL}}(\mathcal{X}, \mathcal{Y}; \theta)$.

In contrast, the proposed TSSDL method uses the following loss function to train a DCNN model:

$$\ell^{\mathrm{TSSDL}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}) = \sum_{i=1}^{N} r_i \cdot \mathrm{CEsoftmax}(\mathbf{W}\mathbf{f}(\mathbf{X}_i; \theta), \tilde{\mathbf{y}}_i), \tag{3}$$

where $\tilde{\mathcal{Y}} = \{\tilde{\mathbf{y}}_i\}_{i=1}^N$ is the estimated set of label vectors for the training set $\mathcal{X} = \{\mathbf{X}_i\}_{i=1}^N$, and each element r_i of $\mathcal{R} = \{r_i\}_{i=1}^N$ is the confidence level for sample $\mathbf{X}_i$, which indicates how reliable is the estimated label vector $\tilde{\mathbf{y}}_i$ of $\mathbf{X}_i$, and is computed in a self-consistent way (to be explained below). If $\mathbf{X}_i \in \mathcal{L}$, $\tilde{\mathbf{y}}_i$ is fixed to its ground-truth label vector $\tilde{\mathbf{y}}_i = \mathbf{y}_i$ throughout the entire training process. For unlabeled training sample $\mathbf{X}_i \in \mathcal{U}$, $\tilde{\mathbf{y}}_i$ is the estimate of its label vector by the current version of the network, and is treated as an optimization variable. As the transductive learning process progresses to its convergence, $\tilde{\mathbf{y}}_i$ gets iterative updates, and converges to the final predicted label vector for $\mathbf{X}_i$. The transductive learning process aims to learn optimal sets of θ^*, $\tilde{\mathcal{Y}}^*$ and $\mathcal{R}^*$ that jointly minimize the loss function:

$$(\tilde{\mathcal{Y}}^*, \theta^*, \mathcal{R}^*) = \arg\min_{\tilde{\mathcal{Y}}, \theta, \mathcal{R}} \ell^{\mathrm{TSSDL}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R})\,. \tag{4}$$

It is noteworthy to point out that the proposed TSSDL method is different from traditional Transductive Semi-Supervised Learning (TSSL) methods in the following two aspects:

(1) Traditional TSSL methods require a fixed feature descriptor $\mathbf{f}(\mathbf{X}_i)$ for each training sample $\mathbf{X}_i$, whereas the proposed TSSDL method keeps learning, and gradually optimizes $\mathbf{f}(\mathbf{X}_i)$ throughout the training process.
(2) Traditional TSSL methods treat every unlabeled sample $\mathbf{X}_i \in \mathcal{U}$ equally, which makes the learning process vulnerable to outliers and uncertain samples. In contract, the proposed TSSDL method introduces the confidence level r_i for each sample $\mathbf{X}_i$ to discount influences from those adverse samples.

We compute the confidence level r_i for $\mathbf{X}_i$ as follows. For each labeled sample $\mathbf{X}_i \in \mathcal{L}$, we always set its confidence level $r_i = 1$. For each unlabeled sample $\mathbf{X}_i \in \mathcal{U}$, we compute r_i based on the intuition that: (i) outliers and highly uncertain samples usually reside in sparsely populated areas in the feature space; and (ii) samples located in densely populated areas are more likely to be assigned correct labels. Let $\{\mathbf{f}_1, \cdots, \mathbf{f}_N\}$ be the learned feature descriptors of $\{\mathbf{X}_1, \cdots, \mathbf{X}_N\}$ outputted by the current version of the DCNN model (i.e., $\mathbf{f}_i = \mathbf{f}(\mathbf{X}_i; \theta)$). We define the proximity value d_i for $\mathbf{X}_i$ as follows:

$$d_i = \sum_{\mathbf{f}_j \in \mathcal{N}(\mathbf{f}_i)} \|\mathbf{f}_i - \mathbf{f}_j\|_2, \tag{5}$$

where $\mathcal{N}(\mathbf{f}_i)$ is the set of k-nearest neighbors (kNN) of $\mathbf{f}_i$. Clearly, a small d_i corresponds to a sample $\mathbf{X}_i$ that is located in a densely populated region, and is more likely to receive the correct label, and vice versa. Therefore, the confidence level r_i of $\mathbf{X}_i$ can be defined as:

$$r_i = 1 - \frac{d_i}{d_{max}}\,,\ d_{max} = \max\{d_1, \cdots, d_N\}\,. \tag{6}$$

Note that each loop when the network parameter set θ is updated, the learned feature descriptor set $\{\mathbf{f}_1, \cdots, \mathbf{f}_N\}$ will be changed. Therefore, we need to recompute the confidence level set $\mathcal{R} = \{r_i\}_{i=1}^N$ using the renewed feature descriptors after each loop of the training process.

3.3 Learning Robust Min-Max Features (RMMF)

Another main component of the proposed TSSDL method is to make the given DCNN model learn the Robust Min-Max Features (RMMF) to further improve the image classification accuracies. This goal is accomplished by adding two regularization terms to the loss function Eq. (3), one for learning the Min-Max Features, and the other for learning the Robust Features. The following part of this section provides detailed descriptions of the two regularization terms.

The Min-Max Feature (MMF) regularization aims to accomplish such properties that in the learned feature space: (i) the distances of images within the same class are minimized; and (ii) the distances of images between different classes are larger than a predefined margin. Based on this statement, we can define the MMF regularization term as follows:

$$R^{\mathrm{MMF}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}) = \\ \sum_{i,j=1}^{N} r_i r_j \big(\|\mathbf{f}_i - \mathbf{f}_j\|^2 \delta(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) - \min(0, \|\mathbf{f}_i - \mathbf{f}_j\|^2 - h)(1 - \delta(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j))\big), \tag{7}$$

where $\delta(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) = 1$ if $\tilde{\mathbf{y}}_i = \tilde{\mathbf{y}}_j$, and 0 otherwise. h is the predefined margin, and all other symbols have the same meanings as in Eq. (3).

As briefly explained in Section 2, the Robust Feature (RF) regularization turns out to be useful for improving image classification accuracies [24, 25, 49]. The main idea is as follows: For each sample $\mathbf{X}_i$, its two perturbed versions $\mathbf{X}_i + \eta_i, \mathbf{X}_i + \eta_i'$ are generated by adding two different random data perturbations η_i, η_i' to $\mathbf{X}_i$, and we require that the difference between the feature descriptors of $\mathbf{X}_i + \eta_i, \mathbf{X}_i + \eta_i'$ be minimized. Translating this statement into equation, we have:

$$R^{\mathrm{RF}}(\mathcal{X}; \theta) = \sum_{i=1}^{N} \|\mathbf{f}(\mathbf{X}_i + \eta_i; \theta) - \mathbf{f}(\mathbf{X}_i + \eta_i'; \theta)\|^2\,. \tag{8}$$

Note that since $\mathbf{X}_i + \eta_i, \mathbf{X}_j + \eta_i'$ are the two perturbed versions of $\mathbf{X}_i$, as they pass forward along the DCNN with dropout enabled, the dropout for $\mathbf{X}_i + \eta_i$ is mostly different from the dropout for $\mathbf{X}_i + \eta_i'$. Thus the perturbations include

data augmentation and dropout within the network. This is a stronger effect and leads to a stronger robustness.

In summary, by combining the above two regularization terms, we enforce the DCNN model to learn the Robust Min-Max Features:

$$R^{\mathrm{RMMF}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}) = \lambda_1 R^{\mathrm{MMF}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}) + \lambda_2 R^{\mathrm{RF}}(\mathcal{X}; \theta)\,, \tag{9}$$

where λ_1 and λ_2 are the parameters to control the tradeoff between the two terms. The overall loss function of the proposed TSSDL method takes the following form:

$$\ell^{\mathrm{TSSDL}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}) = \sum_{i=1}^{N} r_i \cdot \mathrm{CEsoftmax}(\mathbf{W}\mathbf{f}(\mathbf{X}_i; \theta), \tilde{\mathbf{y}}_i) + R^{\mathrm{RMMF}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}). \tag{10}$$

3.4 Optimization of TSSDL Method

Next, we describe the optimization of the loss function Eq. (10). The entire training algorithm for the proposed TSSDL method is shown in Algorithm 3.1. In the following, we provide details of the Steps 4 and 5 in Algorithm 3.1,

Step 4. Fixing the network parameter θ and the confidence level set $\mathcal{R}$, we wish to obtain the optimal label vector set $\tilde{\mathcal{Y}}$. In fact, we only need to obtain the optimal solution of $\tilde{\mathbf{y}}_i$ for each unlabeled sample $\mathbf{X}_i$, $(i = L+1, \cdots, N)$. For simplicity without any confusion, we use $\mathbf{y}_i$ instead of $\tilde{\mathbf{y}}_i$. Let $\mathbf{p}_i = [p_{1i}, p_{2i}, \cdots, p_{Ki}]$ be the prediction score vector of image $\mathbf{X}_i$ (i.e., $\mathbf{p}_i$ is the softmax normalization of the output of the DCNN model's last layer), where p_{ji} represents the prediction score of image $\mathbf{X}_i$ on the j^{th} class.

The relevant term in Eq. (10) is the first term, which can be expressed as:

$$\sum_{\mathbf{X}_i \in \mathcal{U}} r_i \cdot \mathrm{CEsoftmax}(\mathbf{W}\mathbf{f}(\mathbf{X}_i; \theta), \tilde{\mathbf{y}}_i) = -\log \prod_{\mathbf{X}_i \in \mathcal{U}} \left(p_{1i}^{y_i^1} p_{2i}^{y_i^2} \cdots p_{Ki}^{y_i^K} \right)^{r_i}, \tag{11}$$

where we express $\mathbf{y}_i = [y_i^1, \cdots, y_i^K]$ in the component form. Clearly, different data instances i decouple, thus the optimization for Eq. (11) becomes $|\mathcal{U}|$ independent subproblems:

$$\max_{\mathbf{y}_i} \ \log \left(p_{1i}^{y_i^1} p_{2i}^{y_i^2} \cdots p_{Ki}^{y_i^K} \right)^{r_i} \tag{12}$$

Since $r_i \geq 0$, this optimization becomes:

$$\max_{y_i^1 \cdots y_i^K} \ \sum_{k=1}^{K} y_i^k \log(p_{ki}) \tag{13}$$

$$\text{subject to} \ \sum_{k=1}^{K} y_i^k = 1, \ y_i^k \geq 0 \tag{14}$$

Algorithm 3.1 Training algorithm for our proposed TSSDL method.

Input: Training set $\mathcal{D} = \mathcal{L} \bigcup \mathcal{U}$, parameters λ_1, λ_2, the number of loops $Tmax$ (we set $Tmax = 3$).

Output: Network parameter set θ of the DCNN.

1: Train the DCNN on labeled data $\mathcal{L}$ using a supervised learning way.
2: **for** $loop = 1$ to $Tmax$ **do**
3: Fixing θ, update the confidence level set $\mathcal{R} = \{r_i\}_{i=1}^N$ using Eq. (6).
4: Fixing θ and $\mathcal{R}$, optimize $\tilde{\mathcal{Y}}$.
5: Fixing $\mathcal{R}$ and $\tilde{\mathcal{Y}}$, optimize θ on the entire training set $\mathcal{D}$ with the loss function Eq. (10) using mini-batch based stochastic gradient descent from scratch, until the trained DCNN has converged.
6: **end for**

The optimal solution to this problem is given by $y_i^s = 1$ if $s = \arg\max_k p_{ki}$, $y_i^s = 0$ otherwise ($s = 1, \cdots, K$). Thus the optimal solution to Eq.(12) is given by

$$\begin{cases} y_i^s = 1 & \text{if } \ s = \arg\max_k p_{ki} \\ y_i^s = 0 & \text{otherwise} \end{cases} \tag{15}$$

In summary, the optimal solution to Step 4 is given by Eq.(15).

Step 5. This is the DCNN back-propagation (BP) algorithm using stochastic gradient. The gradient of first term of Eq. (10) is computed in standard way. The gradient of R^{RF} is readily computed in terms of $\frac{\partial \mathbf{f}_i}{\partial \theta}$. Gradient of R^{MMF} can be easily computed numerically: $\frac{\partial R^{\text{MMF}}}{\partial \theta} = \sum_{i=1}^N \frac{\partial R^{\text{MMF}}}{\partial \mathbf{f}_i} \frac{\partial \mathbf{f}_i}{\partial \theta}$ and

$$\frac{\partial R^{\text{MMF}}}{\partial \mathbf{f}_i} = \sum_{j=1}^N r_i r_j 2(\mathbf{f}_i - \mathbf{f}_j)\big[\delta(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) - \psi(\|\mathbf{f}_i - \mathbf{f}_j\|^2 - h)(1 - \delta(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j))\big], \tag{16}$$

where $\psi(a) = 1$ if $a < 0$. Otherwise $\psi(a) = 0$. $\frac{\partial \mathbf{f}_i}{\partial \theta}$ is part of the CNN back-propagation, thus easily handled.

3.5 TSSDL Mean-Teacher (TSSDL-MT) Method

In our experiment, we also implemented a TSSDL variant that is the combination of TSSDL and the Mean Teacher methods (TSSDL-MT in short) [26]. Its loss function is defined as:

$$\ell^{\text{TSSDL-MT}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}) = \sum_{i=1}^N r_i \cdot \text{CEsoftmax}(\mathbf{W}\mathbf{f}(\mathbf{X}_i; \theta), \tilde{\mathbf{y}}_i) +$$

$$\lambda_1 R^{\text{MMF}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R}) + \lambda_2 \sum_{i=1}^N \|\mathbf{f}(\mathbf{X}_i + \eta_i; \theta) - \mathbf{f}(\mathbf{X}_i + \eta_i'; \theta')\|^2, \tag{17}$$

where $\theta_t' = \alpha\theta_{t-1}' + (1-\alpha)\theta_t$, α is an exponential moving average (EMA) parameter.The base model using θ is the student model, and θ' is the teacher model. The optimization of the TSSDL-MT method is similar to that of the TSSDL method, and we abide by the same setting of α as that of [26].

4 Experiments

4.1 Experimental Setups

We evaluate the proposed TSSDL method on two benchmark datasets CIFAR10 [50] and SVHN [51]. We choose these two datasets because many recent SSL methods also used them for performance evaluations, which makes it possible to compare TSSDL with these methods. For fair comparison, we use the same 13-layer D-CNN architecture, perturbations, and hyper-parameters (such as weight decay, learning rate, drop ratio, etc) as the Π model [25] and the Mean Teacher model [26]. We conduct all the experiments using TensorFlow [52], and all the models under comparisons are trained from scratch without pre-training. Based on our experiments, we set the margin $h = 1$, $\lambda_1 = 10^{-3}$, $\lambda_2 = 100$ for the TSSDL model, and $\lambda_2 = 1$ for the TSSDL-MT model.

To further reveal how the two regularization terms R^{MMF}, R^{RF} contribute to the performance improvement, in addition to the TSSDL and TSSDL-MT, we also implement the following four variants of the proposed TSSDL method:

- TDCNN: The network is trained without using the two regularization terms R^{MMF} and R^{RF}.
- TMMF: The network is trained without using the regularization term R^{RF}.
- TRF: The network is trained without using the regularization term R^{MMF}.
- Fully Supervised: The network is trained using the standard fully supervised training algorithm with the loss function $\ell^{\mathrm{SL}}(\mathcal{X}, \mathcal{Y}; \theta)$ in Eq. (2) instead of $\ell^{\mathrm{TSSDL}}(\mathcal{X}, \tilde{\mathcal{Y}}; \theta, \mathcal{R})$ in Eq. (3).

4.2 Datasets

The details of the CIFAR10 [50] and SVHN [51] datasets, and their usages for the experimental evaluations are explained as follows.

CIFAR10 dataset. It contains 10 classes of 60000 natural images, which are split into 50000, 10000 images to form the training and test sets, respectively. All the images are 32×32 RGB images. We followed the same training and testing protocols as [26] in the experimental evaluations, where 1000, 2000, 4000, and all the 50000 images (i.e., 100, 200, 400, and 5000 samples per class) are selected from the training set as the labeled training samples, respectively, and the remaining samples in the training set are used as the unlabeled training samples.

SVHN dataset. It contains 73257 training and 26032 test images. All images are 32×32 RGB images. In each image, there can be multiple digits, but the task is to recognize the digit in the image center. Again, following the same training and testing protocols as [26], we select 250, 500, 1000 (i.e., 25, 50, 100 samples per class), and all the 73257 images from the training set as the labeled training samples, respectively, and use the remaining samples in the training set as the unlabeled ones in the experimental evaluations.

Table 1. Top-1 error rates (%) on CIFAR10 test set, averaged over 10 runs.

Method	No. of labeled samples (L)			
	1000	2000	4000	50000 (all)
Ladder Networks [47]	−−	−−	20.40 ± 0.47	−−
Entropy [37]	−−	−−	20.3 ± 0.5	−−
GAN [42]	21.83 ± 2.01	19.61 ± 2.09	18.63 ± 2.32	−−
Sajjadi *et al.* [46]	−−	−−	11.29 ± 0.24	−−
VAT [27]	−−	−−	11.36	5.81
Π model [25]	−−	−−	12.36 ± 0.31	5.56 ± 0.10
TempEns [25]	−−	−−	12.16 ± 0.24	5.60 ± 0.10
Mean Teacher [26]	21.55 ± 1.48	15.73 ± 0.31	12.31 ± 0.28	5.94 ± 0.15
Fully Supervised [26]	46.43 ± 1.21	33.94 ± 0.73	20.66 ± 0.57	6.45 ± 0.15
TDCNN	32.67 ± 1.93	22.99 ± 0.79	16.17 ± 0.37	6.45 ± 0.15
TMMF	26.73 ± 1.11	17.48 ± 0.66	13.11 ± 0.33	5.80 ± 0.17
TRF	27.36 ± 1.30	18.02 ± 0.60	13.30 ± 0.27	6.16 ± 0.11
TSSDL	21.13 ± 1.17	14.65 ± 0.33	10.90 ± 0.23	5.20 ± 0.14
TSSDL-MT	$\mathbf{18.41 \pm 0.92}$	$\mathbf{13.54 \pm 0.32}$	$\mathbf{9.30 \pm 0.55}$	$\mathbf{5.19 \pm 0.14}$

4.3 Comparison to state-of-the-art Methods

Tables 1 and 2 report the experimental results of all the evaluated methods on the CIFAR10 and SVHN test sets, respectively. The four data columns in the righthand side of Table 1 correspond to the top-1 error rates of the respective models trained using 1000, 2000, 4000, and all the labeled training samples in the CIFAR10 training set, respectively, while the four data columns in Table 2 correspond to the top-1 error rates of the respective models trained using 250, 500, 1000, and all the labeled training samples in the SVHN training set, respectively. All the results are averaged over 10 runs with different seeds for data splits. The two tables also include the results of the state-of-the-art SSL methods described in Section 2. We use the same abbreviations for these SSL methods as in Section 2 to report their results. The experimental results in the two tables can be summarized as follows.

- TSSDL and all its variants dramatically outperform the "Fully Supervised" counterpart, proving that the proposed TSSDL method is effective for using the unlabeled samples to improve image classification accuracies of the DCNN model.
- Compared with the baseline TDCNN, adding either the MMF or the RF regularization terms to the loss function can remarkably reduce the error rates on the two test sets.
- TMMF and TRF produce compatible error rates. Combining them together (i.e., TSSDL) achieves the second best performance accuracies on CIFAR10 and compatible results on SVHN with the Mean Teacher method. This is a strong evidence that the proposed MMRF regularization is quite effective for making the DCNN model learn better feature descriptors for the image classification task.

Table 2. Top-1 error rates (%) on SVHN test set, averaged over 10 runs.

Method	No. of labeled samples (L)			
	250	500	1000	73257 (all)
GAN [42]	−−	18.44 ± 4.8	8.11 ± 1.3	−−
VAT [27]	−−	−−	5.42	−−
Haeusser *et al.* [41]	−−	6.25 ± 0.32	5.14 ± 0.17	3.09 ± 0.06
Π model [25]	−−	6.65 ± 0.53	4.82 ± 0.17	2.54 ± 0.04
TempEns [25]	−−	5.12 ± 0.13	4.42 ± 0.16	2.74 ± 0.06
Mean Teacher [26]	4.35 ± 0.50	4.18 ± 0.27	3.95 ± 0.19	2.50 ± 0.05
Fully Supervised [28]	42.65 ± 2.68	22.08 ± 0.73	14.46 ± 0.71	2.81 ± 0.07
TDCNN	22.90 ± 1.91	13.79 ± 1.24	8.77 ± 0.82	2.81 ± 0.07
TMMF	12.99 ± 1.02	7.23 ± 0.76	4.25 ± 0.33	2.30 ± 0.06
TRF	9.93 ± 1.15	6.83 ± 0.66	4.95 ± 0.26	2.65 ± 0.04
TSSDL	5.02 ± 0.26	4.32 ± 0.30	3.80 ± 0.27	2.42 ± 0.05
TSSDL-MT	$\mathbf{4.09 \pm 0.42}$	$\mathbf{3.90 \pm 0.27}$	$\mathbf{3.35 \pm 0.27}$	$\mathbf{2.10 \pm 0.07}$

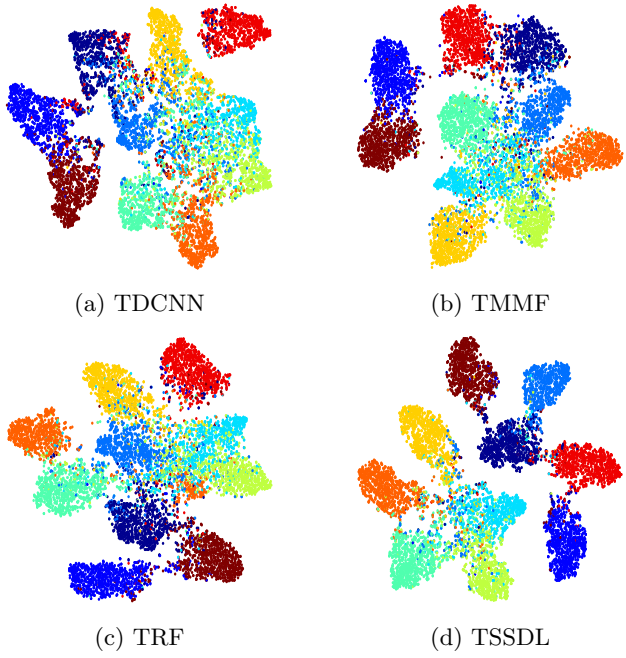
Table 3. Top-1 error rates (%) on SVHN test set with extra unlabeled training data, averaged over 10 runs. The number of labeled samples is 500. $N_u = 73257 - 500$ is the number of unlabeled samples in the original training set.

Method	No. of unlabeled samples		
	N_u	$N_u + 100000$	$N_u + 500000$
TSSDL-MT	3.90 ± 0.27	2.96 ± 0.22	2.27 ± 0.09

– The TSSDL-MT method that combines TSSDL with the Mean Teacher method remarkably outperforms all the methods under comparisons.

4.4 Increasing Extra Unlabeled Samples for SVHN

Tables 1 and 2 indicate that the proposed TSSDL method can effectively use massive unlabeled training samples to improve image classification accuracies. Here, we make an additional experiment on SVHN to test whether TSSDL can achieve a better image classification accuracy by using more unlabeled training samples. Apart from the original training set, SVHN also contains an extra set of 531131 images. Similar to [26], we pick only 500 images from the original training set as the labeled samples, and use the rest of the original training set together with the entire extra set to form the pool of unlabeled samples. Let $N_u = 73257 - 500$ be the number of unlabeled samples in the original training set. We run experiments with TSSDL-MT by using 0, 100000, and 500000 extra unlabeled samples (plus 500 labeled samples and N_u unlabeled samples in the primary training set), respectively. Table 3 shows that: (i) TSSDL-MT does further improve the classification accuracy by using extra unlabeled samples; and (ii) the degree of improvement is positively related to the number of extra unlabeled samples.



(a) TDCNN (b) TMMF

(c) TRF (d) TSSDL

Fig. 1. Feature visualization of the CIFAR10 test set with (a) TDCNN, (b) TMMF, (c) TRF and (d) TSSDL, respectively. Each dot in the figure corresponds to an image, and different colors represent different classes.

4.5 Feature Visualization

We utilize t-SNE [53] to visualize the learned feature descriptors extracted by the TDCNN, TMMF, TRF and TSSDL methods ($L = 4000$) on the CIFAR10 test set, respectively (see Figure 1). It can be observed that: (i) feature descriptors learned by TMMF are better than that of TDCNN, and (ii) feature descriptors learned by TSSDL are the best in terms of between-class separability and within-class compactness. This agrees with the evaluation results in Tables 1 and 2, serving as another evidence for the effectiveness of the proposed TSSDL method.

4.6 Ablation Study

Table 4 lists the ablation study results on CIFAR10 and SVHN. It shows that TSSDL without the confidence levels yields much worse results, especially when the number of labeled samples (L) is small. Indeed, transductive learning together with confidences are an inseparable combination for DCNN application. The proposed TSSDL method is not a simple application of transductive learning to DCNNs. Transductive learning relies on computing distances between feature vectors. Traditional transductive learning requires fixed, high-quality features, while with DCNN, features are of low quality at the beginning and evolve over

Table 4. Top-1 error rates (%) on CIFAR10 and SVHN test sets.

Method	L for CIFAR10			L for SVHN		
	1000	2000	4000	250	500	1000
Fully Supervised	46.43	33.94	20.66	42.65	22.08	14.46
TSSDL-No confidence	50.38	35.67	18.10	46.74	19.10	10.10
TSSDL	21.13	14.65	10.90	5.02	4.32	3.80

time during training. We introduce confidence estimates on inferred labels to avoid the training process from getting into a wrong direction. Table 4 shows that the confidences play an essential role for successful transductive learning application to DCNNs.

5 Conclusions

In this paper, we propose Transductive Semi-Supervised Deep Learning (TSSDL) method that is effective for training Deep Convolutional Neural Network (DCNN) models. The method applies transductive learning principle to DCNN training, introduces confidence levels on unlabeled data samples to overcome unreliable label estimates on outliers and uncertain samples, and uses the Min-Max Feature (MMF) regularization that encourages DCNN to learn features of same-class images be close, and features of different classes be separated by a predefined margin. Extensive experiments on the benchmark datasets CIFAR10 and SVHN have shown that the DCNN model trained by the proposed TSSDL method can produce image classification accuracies that are compatible to the state-of-the-art SSL methods, and that combining TSSDL with the Mean Teacher method can produce the best classification accuracies on the two benchmark datasets. Experiments (Tables 1, 2) show that as the number of labeled data increase, TSSDL performance improves consistently. Experiments (Table 3) also show that as number of unlabeled data increases while the number of labeled data is fixed, TSSDL performance improves consistently. Feature visualizations (Figure 1) show that the Min-Max feature regularization enforces TSSDL to learn feature descriptors with better between-class separability and within-class compactness, thus better discriminative ability.

Acknowledgments. This work is supported by National Basic Research Program of China (973 Program) under Grant No. 2015CB351705, and the National Natural Science Foundation of China (NSFC) under Grant No. 61332018.

References

1. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2009) 1794–1801
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. (2012) 1097–1105
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 770–778
4. Wu, S., Ji, Q., Wang, S., Wong, H.S., Yu, Z., Xu, Y.: Semi-supervised image classification with self-paced cross-task networks. IEEE Transactions on Multimedia **20**(4) (2018) 851–865
5. Shi, W., Gong, Y., Wang, J.: Improving cnn performance with min-max objective. In: Proceedings of the International Joint Conference on Artificial Intelligence. (2016) 2004–2010
6. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1440–1448
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. (2015) 91–99
8. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 815–823
9. Sun, Y., Liang, D., Wang, X., Tang, X.: Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873 (2015)
10. Shi, W., Gong, Y., Tao, X., Wang, J., Zheng, N.: Improving cnn performance accuracies with min-max objective. IEEE Transactions on Neural Networks and Learning Systems **29**(7) (2018) 2872–2885
11. Zhao, F., Huang, Y., Wang, L., Tan, T.: Deep semantic ranking based hashing for multi-label image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1556–1564
12. Yu, M., Liu, L., Shao, L.: Binary set embedding for cross-modal retrieval. IEEE transactions on neural networks and learning systems **28**(12) (2017) 2899–2910
13. Liu, Q., Liu, G., Li, L., Yuan, X.T., Wang, M., Liu, W.: Reversed spectral hashing. IEEE transactions on neural networks and learning systems **29**(6) (2018) 2441–2449
14. Huang, L.K., Yang, Q., Zheng, W.S.: Online hashing. IEEE transactions on neural networks and learning systems **29**(6) (2018) 2309–2322
15. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University (2002)
16. Whitney, M., Sarkar, A.: Bootstrapping via graph propagation. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. (2012) 620–628
17. Gong, C., Tao, D., Liu, W., Liu, L., Yang, J.: Label propagation via teaching-to-learn and learning-to-teach. IEEE transactions on neural networks and learning systems **28**(6) (2017) 1452–1465

18. Pei, X., Chen, C., Guan, Y.: Joint sparse representation and embedding propagation learning: a framework for graph-based semisupervised learning. IEEE transactions on neural networks and learning systems **28**(12) (2017) 2949–2960
19. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the International Conference on Machine Learning. Volume 99. (1999) 200–209
20. Joachims, T.: Transductive learning via spectral graph partitioning. In: Proceedings of the International Conference on Machine Learning. (2003) 290–297
21. Zhang, Y.M., Huang, K., Geng, G.G., Liu, C.L.: Mtc: A fast and robust graph-based transductive learning method. IEEE transactions on neural networks and learning systems **26**(9) (2015) 1979–1991
22. Wang, Z., Zhu, X., Adeli, E., Zhu, Y., Zu, C., Nie, F., Shen, D., Wu, G.: Progressive graph-based transductive learning for multi-modal classification of brain disorder disease. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. (2016) 291–299
23. Görnitz, N., Lima, L.A., Varella, L.E., Müller, K.R., Nakajima, S.: Transductive regression for data with latent dependence structure. IEEE Transactions on Neural Networks and Learning Systems **29**(7) (2018) 2743–2756
24. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems. (2016) 1163–1171
25. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
26. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems. (2017) 1195–1204
27. Miyato, T., Maeda, S.i., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. arXiv preprint arXiv:1704.03976 (2017)
28. Luo, Y., Zhu, J., Li, M., Ren, Y., Zhang, B.: Smooth neighbors on teacher graphs for semi-supervised learning. arXiv preprint arXiv:1711.00258 (2017)
29. de Sa, V.R.: Learning classification with unlabeled data. In: Advances in Neural Information Processing Systems. (1994) 112–119
30. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. (1998) 92–100
31. Cozman, F.G., Cohen, I., Cirelo, M.C.: Semi-supervised learning of mixture models. In: Proceedings of the International Conference on Machine Learning. (2003) 99–106
32. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: Application of Computer Vision. (2005) 29–36
33. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems. (2010) 253–256
34. Jarrett, K., Kavukcuoglu, K., LeCun, Y., et al.: What is the best multi-stage architecture for object recognition? In: Proceedings of the IEEE International Conference on Computer Vision. (2009) 2146–2153
35. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 37–45

36. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1422–1430
37. Hoffer, E., Ailon, N.: Semi-supervised deep learning by metric embedding. arXiv preprint arXiv:1611.01449 (2016)
38. Weston, J., Ratle, F., Collobert, R.: Deep learning via semi-supervised embedding. In: Proceedings of the international conference on Machine learning. (2008) 1168–1175
39. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems. (2014) 3581–3589
40. Abbasnejad, M.E., Dick, A., van den Hengel, A.: Infinite variational autoencoder for semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 781–790
41. Haeusser, P., Mordvintsev, A., Cremers, D.: Learning by association-a versatile semi-supervised training method for neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 89–98
42. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. (2016) 2234–2242
43. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. arXiv preprint arXiv:1511.06390 (2015)
44. Chongxuan, L., Xu, T., Zhu, J., Zhang, B.: Triple generative adversarial nets. In: Advances in Neural Information Processing Systems. (2017) 4091–4101
45. Dai, Z., Yang, Z., Yang, F., Cohen, W.W., Salakhutdinov, R.R.: Good semi-supervised learning that requires a bad gan. In: Advances in Neural Information Processing Systems. (2017) 6513–6523
46. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems. (2016) 1163–1171
47. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems. (2015) 3546–3554
48. Valpola, H.: From neural pca to deep unsupervised learning. In: Advances in Independent Component Analysis and Learning Machines. (2015) 143–171
49. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Advances in Neural Information Processing Systems. (2014) 766–774
50. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, University of Toronto (2009)
51. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: Neural Information Processing Systems (NIPS) workshop on deep learning and unsupervised feature learning. Volume 2011. (2011) 5
52. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
53. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(Nov) (2008) 2579–2605