ADVERSARIAL DROPOUT REGULARIZATION

Kuniaki Saito,Yoshitaka Ushiku,Tatsuya Harada,Kate Saenko

Abstract

The papaer presents a method for transferring neural representations from label-rich source domains to unlabeled target domains. Recent adversarial methods proposed for this task learn to align features across domains by fooling a special domain critic network. However, these methods do not account for preserving class boundaries and may produce ambiguous features near these boundaries, impacting classification accuracy. The paper introduces Adversarial Dropout Regularization (ADR), a novel approach that encourages the generation of more discriminative features for the target domain. Instead of a traditional critic, ADR uses dropout on the classifier network to detect non-discriminative features, prompting the generator to avoid these regions in the feature space. The paper applys ADR to unsupervised domain adaptation for image classification and semantic segmentation tasks and show significant improvements over the state of the art. Additionally, we demonstrate that ADR can be beneficial in training Generative Adversarial Networks for semi-supervised learning..

Keywords: domain adaptation, adversarial learning.

1 Introduction

Transferring knowledge learned by deep neural networks on label-rich domains to new target domains is a challenging problem, especially when the source and target input distributions have different characteristics. For example, deep models trained on simulated driving images from games struggle to transfer to real-world domains without labeled data for fine-tuning, necessitating unsupervised domain adaptation to improve the source model.

Recent unsupervised domain adaptation methods use adversarial learning to reduce the discrepancy between source and target features. These methods typically involve a feature encoder and classifier, along with a separate domain classifier (critic) that labels features as either source- or target-domain. The encoder is trained to maximize the critic's errors, thereby aligning features across domains.

However, this approach has limitations because the critic does not consider class information and may lead to the generation of features that are not discriminative, particularly near class boundaries. To address this, we propose that the adaptation model should consider class decision boundaries while aligning features across domains.

2 Related works

2.1 Domain Adaptation

The field of unsupervised domain adaptation (UDA) for visual data focuses on the alignment of feature distributions between source and target domains, a concept supported by theoretical evidence that suggests minimizing domain divergence can reduce the upper bound of target domain error [1]. Several deep learning techniques have adopted this strategy, employing distribution matching at various hidden layers within networks like CNNs [2], [3], [4]. These methods generally do not account for the interaction between the network's decision boundary and the distribution of target features, an aspect our work considers.

2.2 Low-density Separation

The principle of low-density separation, often used in semi-supervised learning (SSL), places the decision boundary in regions with few unlabeled samples, with the aim of enhancing discriminative feature representation [5], [6]. Our approach extends this principle to deep domain adaptation, and it is akin to entropy minimization strategies in SSL [7] [8] incorporated entropy minimization to gauge the distance of samples from decision boundaries. In contrast, our method achieves low-density separation by perturbing the decision boundary and detecting target samples affected by such perturbations.

2.3 Dropout

Dropout is a well-known regularization technique that aims to reduce overfitting in neural networks by randomly omitting units during training, which effectively samples from numerous thinned network architectures [9]. At inference time, the network utilizes all neurons, ensuring that the network remains robust to input noise. Our method employs dropout in an adversarial manner, transforming the classifier into a critic that is sensitive to noise, which is a novel application compared to traditional dropout uses.

3 Method

3.1 Overview

ADR overview Figure 1: The method of ADR uses dropout to generate two classfier C1 and C2,dropping different nodes each time and obtaining two different output vectors denoted as $C_1(G(x_t))$, $C_2(G(x_t))$.Then C should maximize the sensitivity for target samples to detect the samples near the boundary.G should learn to minimize the sensitivity to move target samples away from the boundary.

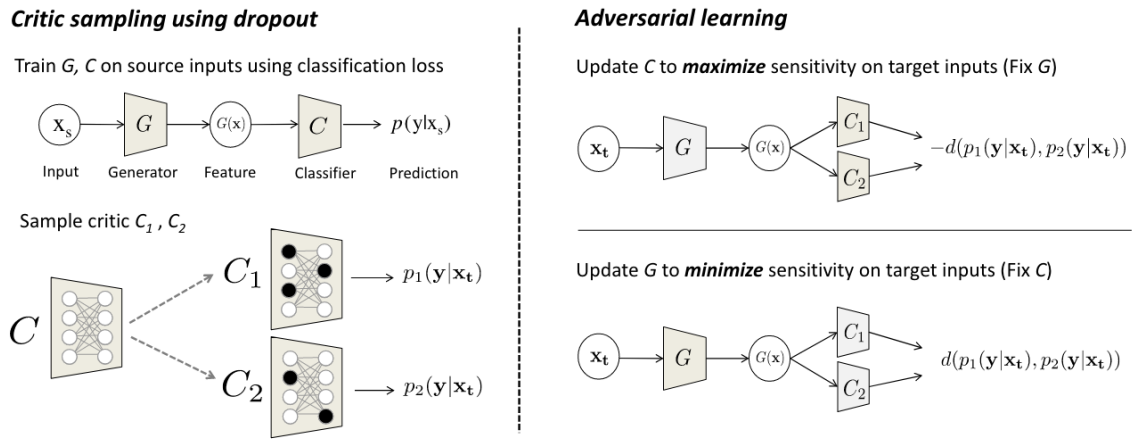

Figure 1. Overview of the method

3.2 Training Procedure

To measure the sensitivity d(p1, p2) between the two obtained probabilistic outputs, we use the symmetric kullback leibler (KL) divergence. Formally, the divergence is calculated as

$$d(p_1, p_2) = \frac{1}{2}(D_{KL}(p_1|p_2) + D_{KL}(p_2|p_1)) \tag{1}$$

where KL divergence between p and q is denoted as $D_{kl}(p|q)$. Step 1, in this step, C is trained as a classifier. C and G have to classify source samples correctly to obtain discriminative features. Thus, we update both networks' parameters based on the following standard classification loss. Given source labels y_s and samples x_s, the objective in this step is

$$\min_{G,C} L(X_s, Y_s) = -\mathbb{E}(x_s, y_s) \sim (X_s, Y_s) \sum k = 1^K \mathbb{1}[k = y_s] \log p(y|x_s) \tag{2}$$

Step 2, in this step, C is trained as a critic to detect target samples near the boundary. Two classifiers are sampled from C for each target sample using dropout twice to obtain p1 and p2. Then, C's parameters are updated to maximize the sensitivity as measured by Eq. 1. Since C should learn discriminative features for source samples, in addition to the sensitivity term, we add Eq. 2. We experimentally confirmed that this term is essential to obtain good performance.

$$\min_{C} L(X_s, Y_s) - L_{adv}(X_t) \tag{3}$$

$$L_{adv}(X_t) = E_{x_t \sim X_t}[d(p_1, p_2)] \tag{4}$$

Step 3, in order to obtain representations where target samples are placed far from the decision boundary, G is trained to minimize sensitivity. Here we do not add the categorical loss for source samples as in Step 2, as the generator is able to obtain discriminative features without it. We also assume that target samples should be uniformly distributed among the possible K classes and add a corresponding conditional entropy term to the generator objective.

$$\min_{G} L_{adv}(X_t) + E_{x_t \sim X_t} \sum_{k=1}^{K} p(y = k|x_t) \log p(y|x_t) \tag{5}$$

We update the parameters of C and G in every step following the defined objectives. We experimentally found it beneficial to repeat Step 3 n times.

4 Implementation details

4.1 Comparing with the released source codes

However, the code published by the authors has only been validated on a few datasets. We have applied the ADR algorithm to more data sets and more scenarios.the code published by the authors is `https://github.com/mil-tokyo/adr_da`

We verified the ADR algorithm on different data sets, which is consistent with the experimental results in the original article.

4.2 Experimental environment setup

We use anaconda to control our environment,the version of anaconda is 4.10.1. The version of pytorch is 2.1.2. The cuda version is 11.8.

4.3 datasets

We use ADR on Digits Classification. The datasets are MNIST (LeCun et al. (1998)), SVHN (Netzer et al. (2011)) and USPS. USPS is a digit dataset automatically scanned from envelopes by the U.S. Postal Service containing a total of 9,298 16×16 pixel grayscale samples; the images are centered, normalized and show a broad range of font styles.

The MNIST database (Modified National Institute of Standards and Technology database) is a large collection of handwritten digits. It has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger NIST Special Database 3 (digits written by employees of the United States Census Bureau) and Special Database 1 (digits written by high school students) which contain monochrome images of handwritten digits.

Street View House Numbers (SVHN) is a digit classification benchmark dataset that contains 600,000 32×32 RGB images of printed digits (from 0 to 9) cropped from pictures of house number plates. The cropped images are centered in the digit of interest, but nearby digits and other distractors are kept in the image. SVHN has three sets: training, testing sets and an extra set with 530,000 images that are less difficult and can be used for helping with the training process.

USPS is a digit dataset automatically scanned from envelopes by the U.S. Postal Service containing a total of 9,298 16×16 pixel grayscale samples; the images are centered, normalized and show a broad range of font styles.

5 Results and analysis

METHOD	SVHN to MNIST	USPS to MNIST	MNIST to USPS
Source Only	67.1	68.1	77.0
ATDA (Saito et al. (2017))	86.2†	-	-
DANN (Ganin & Lempitsky (2014))	73.9	73.0±2.0	77.1±1.8
DoC (Tzeng et al. (2014))	68.1±0.3	66.5±3.3	79.1±0.5
ADDA (Tzeng et al. (2017))	76.0±1.8	90.1±0.8	89.4±0.2
CoGAN (Liu & Tuzel (2016))	did not converge	89.1±0.8	91.2±0.8
DTN (Taigman et al. (2016))	84.7	-	-
ENT	93.7±2.05	89.2±3.07	89.6±1.44
Ours	**94.1**±1.37	**91.5**±3.61	**91.3**±0.65

Figure 2. Experimental results

Results in Table 1 are from ADR paper . We reproduce them and the result is similar.Results in Table 1 demonstrate that ADR obtains better performance than existing methods. In particular, on the challenging adaptation task from SVHN to MNIST, our method achieves much better accuracy than previously reported.

6 Conclusion and future work

The paper introduced a novel approach for aligning feature distributions called Adversarial Dropout Regularization, which learns to generate discriminative features for the target domain. The method consists of a critic network that can detect samples near the boundary and a feature generator that fools the critic. Our approach is general, applies to a variety of tasks, and does not require domain labels. In extensive domain adaptation experiments, our method outperformed baseline methods, including entropy minimization, and achieved state-of-the-art results on three datasets. It also proved effective when applied to Generative Adversarial Networks for semi-supervised learning.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[2] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

[4] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[5] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics*, pages 57–64. PMLR, 2005.

[6] Thorsten Joachims et al. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pages 200–209, 1999.

[7] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

[8] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.