Mixup 정규화를 활용하여 적대적 도메인 적응 향상

칼리나 바야르치멕[O], 조영복[*]
[O]대전대학교 정보보안학과,
[*]대전대학교 정보보안학과
e-mail: bayarchimegkalina@gmail.com[O]

Utilizing Mixup Regularization to improve Adversarial Domain Adaptation

Kalina Bayarchimeg[O], Youngbok-Cho[*]
[O]Dept. of Information Security, Daejeon University,
[*]Dept. of Information Security, Daejeon University

● 요 약 ●

비지도형 도메인 적응(UDA)에 대한 최근 연구는 도메인 적응에 대한 설명 및 전이 가능한 특징을 풀어내기 위해 적대적 학습에 의존한다. 그러나 기존 방법에는 대상 도메인의 클래스 인식(class-aware) 정보를 고려하지 않고는 잠재 공간의 구별 가능성을 완전히 보장할 수 없다는 것과 소스 및 대상 도메인의 샘플만으로는 잠재 공간에서 도메인 불변(domain- invariant) 특성을 추출하기에 부족하다는 두 가지 문제가 있다고 알려져 있다. 본 논문에서는 기존 알려진 UDA의 도메인 적응시 발생되는 문제를 해결하기 위해 Adversarial Discriminative Domain Adaptation(ADDA)에서 mixup을 활용해 신경망의 로버스트네스를 향상시키는 것을 확인하였다.

키워드: 비지도형 도메인 적응, 도메인 불변(domain−invariant), Mixup, 적대적 도메인 적응

I. Introduction

ADDA은 Unsupervised Feature-Level Domain Adaptation 중 하나이다. UDA 모델의 최근 발전은 주로 심층 신경망을 기반으로 하며, 적대적 학습을 사용하여 도메인 전반에 걸쳐 도메인 불변 특징을 학습하는 데 중점을 두는 추세이다. 적대적 도메인 적응 모델은 생성자가 도메인 판별자를 혼동하도록 학습되는 미니맥스 게임을 플레이하여 도메인 전반에 걸쳐 구별 및 도메인 불변 특징을 학습한다. ADDA 방법은 이미지 분류, 의미론적 분할(semantic segmentation)과 같은 다양한 태스크에서 좋은 성능(잠재력)을 보여주었지만, 이러한 방법에는 두 가지 문제가 있다. 첫째, 도메인 분류자는 특징을 소스 또는 대상으로 구분하려고만 하며 클래스 간의 테스크별 결정 경계는 고려하지 않는다는 것이다. 둘째, 각 도메인의 특성으로 인해 서로 다른 도메인 간의 특성 분포를 완전히 일치시키는 것을 목표로 한다는 것이다. 따라서 본 논문에서는 기존 주어진 문제를 해결하기 위해 Mixup을 활용한 네트워크의 특징을 기반으로 도메인별 특징을 분류할수 있도록 한다. 본 논문의 구성은 2장에서 관련연구로 인터플레이션 기반 정규화와 도메인 적응에 대해 기술하고 3장에서는 제안방식인 ADDA에 대해 기술한다. 4장에서는 실험환경 설정 및 실험결과를 기술하고 마지막으로 5장에서 결론을 기술한다.

II. The Proposed Method

1. Mixup

Mixup은 domain-agnostic 데이터 확장 기술로 제안되었다 [3]. 일반적으로 신경망은 손상된 레이블을 기억하는 경향이 지니고 있다. 이러한 문제를 해결하기 위한 mixup은 네트워크가 특징과 레이블 간의 관계에 대해 과신하지 않도록 서로 다른 클래스의 특징을 서로 결합한다. 레이블에도 동일하게 적용된다. 그리고 컴퓨터 비전, 자연어 처리, 음성 등과 같은 다양한 데이터 양식으로 확장될 수 있다.

$$\tilde{x} = \lambda x_i + (1-\lambda) x_j, \quad x_i, x_j$$
$$\tilde{y} = \lambda y_i + (1-\lambda) y_j, \quad y_i, y_j \qquad (2)$$

수식2는 먼저 원시입력벡터와 이를 기반으로 한 원-핫 라벨 인코딩을 수식으로 표현한 것이다. 즉 람다 값은 [0, 1] 범위의 값이며 베타 분포에서 샘플링된다.

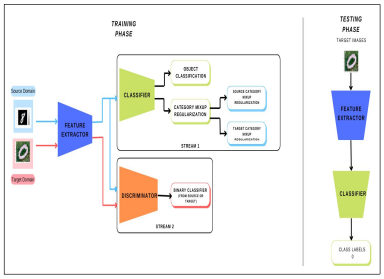

Fig. 1. The Architecture of Proposed method.

그림 1에서는 본 논문에서 제안하는 네트워크 구조를 도식화한 것이다. 그림1와 같이 특징 추출기는 판별 및 도메인 불변 특징을 학습하는 것을 목표로 하고, 도메인 판별기는 샘플링된 특징이 소스 도메인 또는 대상 도메인에서 오는지 여부를 알려주도록 훈련되고, 분류기는 객체 분류를 수행하는 데 사용된다.

III. Experiments and Results

본 논문에서 실험환경은 Generator, Classifier, Discriminator 3 가지 neural network(NN)를 새롭게 설계했다. 우리는 세션에서 클럭 속도가 2.30GHz인 2개의 코어가 있는 Intel CPU가 있는 Google Colab 서버를 사용했다. 메모리를 대해서는 Google Colab은 무료 12GB를 제공한다. 그런데 GPU는 Tesla T4이고 운영체제는 Linux이다. 모든 실험은 PyTorch 플랫폼에서 구현되었으며 Torch 버전은 1.13.0+cu116이었다. 모든 네트워크는 0.0001의 학습률과 Adam 옵티마이저로 처음부터 훈련되었다. 구체적으로 Adam 옵티마이저를 사용하고 learning rate와 batch size를 각각 0.0001, 64로 설정하였다. 50 epochs 후의 결과는 표 1과 같다.

Table 1. Best performance of proposed model

Result	MNIST → MNIST-M	MNIST → USPS	SVHN → MNIST
Test loss	0.0015	0.0021%	0.0111%
Average test accuracy	97%	95%	84%

또한 t-SNE(Maaten and Hinton 2008)를 활용하여 MNIST → MNIST-M, MNIST → USPS 및 SVHN → MNIST 태스크에서 대상 도메인의 특징 분포를 시각화해봤다. 각각 그림 3, 4와 5에서 볼 수 있듯이 Mixup 정규화를 사용하는 ADA에서는 서로 다른 클래스의 특징이 명확하게 구분된다. 그런데 Mixup 정규화 부분을 제외하고 MNIST→MNIST-M, MNIST→USPS 및 SVHN → MNIST 태스크를 추가로 조사했다. 그런데 그 결과 중에서 SVHN → MNIST 태스크가 mixup 정규화가 있는 결과와 비교했을 때 13%의 차이가 났다. 이 것은 MNIST → MNIST-M, MNIST → USPS와 같은 다른 태스크보다 가장 높다.

REFERENCES

[1] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, Y. Bengio, "Manifold Mixup: Better Representations by Interpolating Hidden States", ICML, 2019.

[2] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, "Maximum Classifier Discrepancy for Unsupervised Domain Adaptation", CVPR. pp. 3723-3732, 2018.

[3] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, "mixup: BEYOND EMPIRICAL RISK MINIMIZATION", arXiv:1710.09412, 2018.

[4] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning", arXiv preprint arXiv:1905.02249, 2019.

[5] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, "Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks", arXiv preprint arXiv:1612.05424, CVPR 2017.