Project Plan - Integration of Domain-specific Batch Normalization into Adversarial Neural Networks for Visual Domain Adaptation

Juwon Lee

September 30, 2018

Supervisor: Dr. Luo Ping

Contents

1. **Project Background** 1
2. **Project Objectives** 2
3. **Methodology** 3
4. **Schedule and Milestones** 5
5. **References** 8

1 Project Background

When the distribution of the training dataset significantly differs from that of the test dataset, standard classifiers tend to perform poorly on test data. Such failure to generalize is caused by a problem known as the domain shift. Under supervised learning setting, multiple empirical results have suggested that the extent of domain shift, or "difference" between the distributions of training and test data, are directly proportional to the test error of the model [1]. Without a doubt, in many practical applications in which the training and the test data are not pooled from the same exact domain, domain shift remains a critical barrier to successfully deploying a machine learning model in a real life setting.

Over the past few years, a variety of domain adaptation strategies have been proposed to address data shifts from source (i.e. training) data to target (i.e. test) data. Based on the availability of target data label, domain adaptation can be classified as either supervised or unsupervised. In many real-life settings, presence of labelled target data cannot always be guaranteed, and manual labeling of a new dataset can be an extremely costly task. This lack of diversity and variety in available labeled datasets justify the utility of unsupervised domain adaptation, learning framework in which only the source data are labeled.

In the past, unsupervised domain adaptation has been tackled with different approaches. Wilson et. al [2] lays out and compares the approaches that have been developed thus far, each category summarized as follows:

- Domain-invariant Features Learning
  If the distribution of the features upon some form of transformation are the same between that of the source domain and the target domain, the transformed feature representation is said to be domain-invariant. In theory, a model trained on such domain-invariant features representation is expected to perform as well on target dataset as it does on source dataset. The underlying assumption of this approach is that such representation exists, but sometimes, this assumption is not always true, rendering the approach effective only in limited scenarios.

- Domain Mapping
  Domain mapping approaches attempt to learn the mapping from source domain to target domain so that the datasets from the source domain can be adapted to resemble the distribution of the target domain more closely before being used to train a model. Conditional GANs and Image-to-Image translations are used to achieve such mapping. Limitations of this approach, however, are that the semantic consistency is not always preserved if the learning of mapping is done without class labels.

- Target Discriminative Methods
  Target discriminative methods are driven by the assumption that decision boundaries can be identified by density and that domain adaptation can be

achieved by shifting the decision boundary into regions of low density. Different variations of adversarial training are used to move the decision boundaries in the target domain. However, as these methods heavily rely on the cluster assumption (i.e. assumption that data samples in a cluster belong to the same class), when there is a significant imbalance between clusters, they have suboptimal performance.

- Normalization Statistics
  A few studies have proposed the usage of batch normalization for domain adaptation. The rationale behind such approach is that if the class-related knowledge is stored in the weights of a neural layer and the domain-related knowledge is stored in the batch normalization layer, domain adaptation can be achieved simply by modulating the statistics of the batch normalization layer. Again, the limitations of the approach lie in the differences between source domain and target domain that the batch normalization layer fails to capture.

The overview reveals that despite there being a significant number of existing techniques for unsupervised domain adaptation, each approach has limitations that result in suboptimal performance. The high-level hypothesis I'll be exploring throughout this project is that a strategic combination of two or more domain adaptation techniques would outperform usage of a single technique. More specifically, I'll focus on integration of domain-specific batch normalization into existing adversarial adaptation methods.

2 Project Objectives

The project's main objective is to improve the performance of existing adversarial domain adaptation techniques through integration of domain-specific batch normalization.

Overall, the scope of the project can be summarized as follows:

- Identify previously proposed adversarial domain adaptation techniques through comprehensive literature review. For each in a selected group of algorithms, implement a version with domain-specific batch normalization added in the process.

- Wrap the implementation of adversarial adaptations into an open-source Python module. Provide comprehensive documentation for ease of use by third parties.

- Experimentally validate the efficacy of DSBN integration on each algorithm by testing on standard benchmark datasets. With each algorithm, train the model with and without DSBN and compare the performance.

- Put together datasets of medical images (e.g. chest x-ray and mammograms), each category containing images from different sources. Conduct experiments on medical image datasets to further demonstrate the effectiveness of the approach in practical use.

3 Methodology

3.1 Batch Normalization

Originally developed for the purpose of accelerating deep network training, batch normalization is a technique that reduces internal covariate shift [3]. Internal covariate shift refers to the phenomenon in which distribution of each layer's inputs changes during training, requiring a much smaller learning rate that ultimately slows down training. Batch normalization tackles this problem by normalizing layer inputs. Formally, batch normalization layer transforms a particular activation x_i in a mini-batch B using the following equation

$$\begin{aligned} \hat{x}_i &= \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \\ y_i &= \gamma \hat{x}_i + \beta \end{aligned} \tag{1}$$

where μ_B refers to the mini-batch mean of B, σ_B refers to the mini-batch variance of B, and ε refers to constant added to the mini-batch variance for numerical stability. γ and β refer to learnable parameters that are updated during training.

3.2 Domain-specific Batch Normalization

In [4], Li et. al proposed a technique called Adaptive Batch Normalization, developed based on an assumption that batch normalization statistics store domain-specific knowledge that can be used to further enhance the performance of existing domain adaptation treatments. Simply put, Adaptive batch normalization, or AdaBN for short, re-estimates batch normalization statistics using target samples. Empirical results show that such modulation of batch normalization statistics can successfully improve model performance under unsupervised domain adaptation settings.

In [5], Chang et. al take the assumption discussed in [4] one step further by proposing an algorithm that adapts to both source and target domains. Named domain-specific batch normalization, the proposed method specializes batch normalization layers to source and target domain respectively and allocates domain-specific affine parameters γ and β. In the paper, authors illustrate the supremacy of the approach using multiple benchmark datasets.

3.3 **Adversarial-learning-based Domain Adaptation Methods**

Domain Adversarial Neural Networks (DANN) [6] is an example of domain-invariant features-learning domain adaptation technique that makes use of adversarial learning. In its simplest form, DANN consists of three parts: label predictor, domain classifier, and feature extractor. Parameters of the label predictor are optimized to minimize classification error on the labels of the source data, and parameters of the underlying feature extractor are optimized to minimize the loss of the label classifier while maximizing the loss of the domain classifier. The rationale behind such maximization of domain classifier loss (i.e. adversarial updates for domain classifier) is that it drives learning of domain-invariant features that can "confuse" domain classifier. Such rationale is aligned with the overall theory on domain adaptation that a successful round of domain adaptation yields features characterized by difficult identification on the domain of origin of input observation.

Many variations of adversarial-learning-based domain adaptations have been proposed since [6], and Cycle-consistent Adversarial Domain Adaptation (CyCADA) is one of such variations. While following the aforementioned adversarial approach to mitigate the effects of domain shift, CyCDA further enforces structural and semantic consistency by using a cycle-consistency loss [7]. The additional learning steps include learning the mapping from target to source domain, trained with the same GAN loss as the one used for learning the mapping from source to target domain. Then, to preserve the semantic consistency in adaptation, CyCDA requires the mapping from source to target and subsequently back to source reconstruct the original data. This enforcement is made possible via cycle-consistency loss, which is essentially an L1 penalty on the reconstruction error.

My project will build on the works of [5], [6], and [7], incorporating domain-specific batch normalization into existing adversarial-learning-based domain adaptation techniques including [6] and [7].

3.4 **Performance Evaluation on Benchmark Datasets**

Efficacy of each adjusted technique will be evaluated on two standard benchmark datasets: Office31 and Caltech-Bing.

Office31[8] is a collection of 4652 images from 3 different datasets, each collected from a different domain: Amazon, DSLR, and Webcam. The images are used to train a model that performs image classification task, which outputs the most probable label for the input image out of the 31 classes.

Caltech-Bing is another image dataset used for domain adaptation, and it contains a total of 152337 images in 256 classes [9]. Each image comes from either the domain of Caltech-256 or Bing.

With each dataset, in order to test the efficacy of a domain adaptation technique, different pairs of domains can be constructed in which one domain is used as source domain and the other is used as target domain. For example, a model trained on data from the Amazon domain can be first tested on data from Amazon (source) and then on data from DSLR (target). Likewise, a model trained on data from Caltech-256 can

be tested on Bing data. Classification performance of each algorithm will be tested across all of such pairs.

3.5 Further Performance Evaluation on Medical Datasets

To further validate the efficacy of each approach in practical settings, I'll study their performance on medical datasets in two categories (chest X-rays and mammograms), pulled from different sources. Some potential candidate datasets for chest X-rays include Japanese Society of Radiological Technology CXR datasets [10] and Montgomery and Shenzhen sets [11]. Some potential candidate datasets for mammograms include the Mammographic Image Analysis Society (MIAS) dataset [12] and the Digital Database for Screening Mammography (DDSM) [13].

3.6 Package Implementation

With each dataset, in order to test the efficacy of a domain adaptation technique, different pairs of domains can be constructed in which one domain is used as source domain and the other is used as target domain. For example, a model trained on data from the Amazon domain can be first tested on data from Amazon (source) and then on data from DSLR (target). Likewise, a model trained on data from Caltech-256 can be tested on Bing data. The average drop in classification accuracy across the different pairs will be used as the efficacy measure to evaluate the performance of the domain adaptation technique

4 Project Schedule and Milestones

Major milestones of the project are scheduled as follows:

- **September 2019**
  - **Preliminary literature review**
  a. Study batch normalization[3], adaptive batch normalization[4], and domain-specific batch normalization[5]
  b. Study domain-adversarial neural networks [6]

  - **Phase 1 Deliverables**
  a. Detailed project plan
  b. Project website

- **October 2019**
  - **Implementation of domain-specific batch normalization**
  a. Fork and make adjustments to the code for AdaBN [12] to introduce domain-specificity to batch normalization
  b. Wrap the code in a class of its own to make it easy to plug into other domain adaptation workflows

- **November 2019**
    - **Implementation of domain-adversarial neural networks**

    a. Wrap the code in a class of its own to make it easy to plug into other domain adaptation workflows

    - **Implementation of cycle-consistent domain-adversarial neural networks**

    a. Wrap the code in a class of its own to make it easy to plug into other domain adaptation workflows

    - **Further literature review on more adversarial-learning-based domain adaptations**

    a. Study PixelDA [14] (tentative)
    b. Study CoGAN [15] (tentative)

- **December 2019**

    - **Phase 2 Deliverables**

    a. Preliminary development
    b. Detailed report
    c. First presentation

- **January 2020**

    - **Test the implementations so far on standard benchmark datasets**

    a. Test the implementations on office dataset
    b. Test the implementations on Caltech-Bing dataset

- **February 2020**
    - **Based on the results and the secondary literature reviews, identify promising adversarial domain adaptations to which domain-specific batch normalization can be incorporated. Implement and test the selected approaches.**

- **March 2020**
    - **Test on medical datasets**

    a. Test the implementations on chest x-ray datasets
    b. Test the implementations on mammogram datasets

    - **Write documentation for all implementations and make the implementation code open source**

- **April 2020**
    - **Phase 3 Deliverables**

a. Final presentation
b. Final report

References

1. Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual Domain Adaptation: A survey of recent advances. IEEE Signal Processing Magazine, 32(3), 53–69. doi: 10.1109/msp.2014.2347059
2. Wilson, G., & Cook, D. J. (2018). Survey of Unsupervised Domain Adaptations.
3. Ioffe, S. & Szegedy, C.. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning, in PMLR 37:448-456
4. Li, Y., Wang, N., Shi, J., Hou, X., & Liu, J. (2018). Adaptive Batch Normalization for practical domain adaptation. Pattern Recognition, 80, 109–117. doi: 10.1016/j.patcog.2018.03.005
5. Chang, W.-G., You, T., Seo, S., Kwak, S., & Han, B. (2019). Domain-Specific Batch Normalization for Unsupervised Domain Adaptation.
6. Ganin, Y., & Ustinova, E. (2016). Domain-Adversarial Training of Neural Networks. Journal of Machine Learning Research, 17, 1–35.
7. Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., & Darrell, T. (2018). CyCADA: Cycle-Consistent Adversarial Domain Adaptation. Proceedings of the 35th International Conference on Machine Learning, 80.
8. Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting Visual Category Models to New Domains. Computer Vision – ECCV 2010 Lecture Notes in Computer Science, 213–226. doi: 10.1007/978-3-642-15561-1_16
9. Duan, L., Xu, D., & Chang, S.-F. (2012). Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. 2012 IEEE Conference on Computer Vision and Pattern Recognition. doi: 10.1109/cvpr.2012.6247819
10. Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K.-I., … Doi, K. (2000). Development of a Digital Image Database for Chest Radiographs With and Without a Lung Nodule. American Journal of Roentgenology, 174(1), 71–74. doi: 10.2214/ajr.174.1.1740071
11. Jaeger, S., Candemir, S., Antani, S., Wang, Y.-xing J., Lu, P.-X., & Thoma, G. (2014). Two Public Chest X-Ray Datasets for Computer-Aided Screening of Pulmonary Diseases. Quantitative Imaging in Medicine and Surgery, 4(6), 475.
12. Bowyer, K. W. (1995). Digital Image Database with Gold Standard and Performance Metrics for Mammographic Image Analysis Research. doi: 10.21236/ada300083
13. Daoudi, R., Djemal, K., & Benyettou, A. (2014). Digital Database for Screening Mammography Classification Using Improved Artificial Immune System Approaches. Proceedings of the International Conference on Evolutionary Computation Theory and Applications. doi: 10.5220/0005079602440250
14. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., & Krishnan, D. (2017). Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/cvpr.2017.18
15. Liu, M.-Y., & Tuzel, O. (2016). Coupled Generative Adversarial Networks. Advances in Neural Information Processing Systems, 29.