Using self-supervised learning to improve performance on domain adaptation

Ali Dadsetan
dadsetan.ali@gmail.com

1 Introduction

Given a collection of labeled data that are drawn from a source domain with a specific distribution, the goal of domain adaptation is to make predictions over a target domain from which we have only access to unlabeled data. In other words, the labeled data in the target domain can only be used in the evaluation phase. A simple approach is to fine-tune a neural network using the source dataset and then use this pre-trained model to predict labels on the target domain. However, this approach does not leverage the unlabeled target data points that have been sampled from the target domain.

In this project, we explore the possible benefits of using the unlabeled target dataset in the training process. Specifically, we highlight that by applying a self-supervised learning scheme over the joint target and source datasets (before fine-tuning the network), the performance can increase. This is numerically validated on the VisDA2017 dataset, where we show that the accuracy increases from 51% to 66%.

2 Method

2.1 Dataset

The Visda2017 dataset, (4), consists of a source and target domain, each with images of 12 categories. The source domain consists of synthetic 2D renderings of 3D models generated from different angles and lighting conditions. The target domain consists of real-image photos from the same categories. After fine-tuning a pre-trained Resnet50 architecture over 1500 (training more would reduce the overall performance) samples from the source domain, we achieve a 51% accuracy over the target domain.

The overall design is based on applying the SimCLR algorithm, (2), a self-supervised learning algorithm on the joint dataset consisting of data from both the source and target domains. In order to obtain a better performance, we have made some modifications to the SimCLR algorithm, which we discuss below.

First, there was some modification in the augmentation process. Most notably, for the images of the source domain, rendered images from different angles and with different lighting were treated as different augmentations of the same object. A similiar idea has been used in (1) in the context of medical imaging. We refer to Section 2.2 for more details on data augmentation.

Second, the loss function of the SimCLR algorithm has been changed to consider the labels whenever it is allowed to use them. A similiar approach has been proposed in (3). Remarkably, both methods use labels to find representations that yield close features for images that are in the same category. However, (3) needs to address the limitation of having partial access to the target labels during training. This is essential, because a shared representation of both domains is required for having a good performance over the target domain. More details about the loss function are given in 2.3.

Now the stated hypothesis can be verified. Training the Resnet50 with the modified SimCLR algorithm before fine-tuning improves the accuracy over the target domain to 66%. This improvement in accuracy is the main result of this project so far and shows that self-supervised contrastive learning might be helpful in domain adaptation problems. In the following sub-sections of this section, more details about the previous paragraphs are given.

2.2 Augmentation and more details about dataset

The VisdDA2017 source domain dataset has 150k images, which are 2D renderings of only 1900 different models (i.e., Each model has 80 different renderings.) The target domain has around 50k images in the same categories. The 80 different renderings of the same model are treated as transformations of the same object, so the size of the source dataset effectively will be 1900. For having a balanced number of source and target samples in the self-supervised training dataset, the elements of the source dataset are repeated ten times. Every time an element of this dataset is fetched, a random image of the set of 80 renderings of the object is returned. Furthermore, an additional augmentation in the source domain is done, adding random colors to the original gray-scale image.

2.3 Loss function

Let us start by reviewing the loss function used in the original SimCLR paper. For any given batch with n samples, $x_1, \ldots, x_n$, we could consider the representations after the projection layer, $z_1, \ldots, z_n$, and representations of an augmented copy, $z_{n+1}, \ldots, z_{2n}$. In this setting, for each i, $1 \leq i \leq n$, the SimCLR paper has defined the l_{ij} by

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau}{\sum_{k \neq i} \exp(\text{sim}(z_i, z_k)/\tau)}$$

This is like doing a softmax over the similarities of the pairs of z_i and z_j. So optimizing a loss function like that would make z_i closer to z_j and farther from other z_ks. Then, the SimCLR paper has defined the total loss function of the batch by the summation $L = \sum_{i=1,\ldots,n} \ell_{i,n+i}$, making z_i's more similar to z_{n+i}'s (their augmented counterpart).

Reasons for the Changes made to the original loss function are discussed at the start of this section (2.1). The idea is to change the denominator of the above definition for ℓ_{ij} so that it would penalize distance between objects of the same category more. Also, it should not use the labels of the target domain. This could be done by simply giving different coefficients to different components of the summation in the denominator.

More formally, by defining p_{same} as the penalty coefficient for objects in the same category, $p_{\text{different}}$ as the penalty for objects with different categories, the below definition for ℓ'_{ij} would take care of both of our criteria.

$$\ell'_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k \neq i} p_k \exp(\text{sim}(z_i, z_k)/\tau)}$$

Where p_k is the function

$$p_k = \begin{cases} 1 & x_i \notin \text{source or } x_j \notin \text{source} \\ p_{\text{same}} & y_k = y_i \\ p_{\text{different}} & y_k \neq y_i \end{cases}$$

and y_i is the category (label) of x_i.

In the experiment leading to this report's result, p_{same} was 10 and $p_{\text{different}}$ was 0.1.

2.4 Evaluation

After each epoch of self-supervised training over the joint dataset described in subsection 2.2, the projection layer of the SimCLR network is removed (temporarily) and replaced by a linear layer, with weights of the linear layer randomly initiated. Then after freezing the encoder part, we would train the linear layer over 1% of the source domain labeled data (the so-called fine-tuning) and evaluate the accuracy of the resulting network on all elements of the target domain. This accuracy is reported in Figure as the `adaptation_acc_one_percent`. After that, the projection layer is re-attached and the self-supervised part is continued.

3 Results

We can see in Figure 1 that the lesser the loss function becomes, the more accuracy is achieved on the target domain after fine-tuning. This validates the stated hypothesis in section 1.

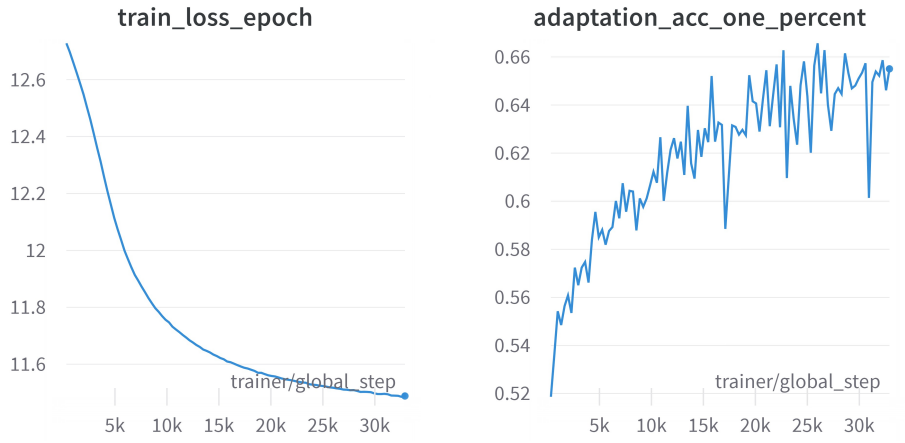

Figure 1: Loss function and accuracy of fine-tuning over the target domain.

Funding Statement

I would like to thank ML collective, as this project was possible because of the computing support provided by them.

References

S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3478–3488, 2021.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge, 2017.