Machine Learning Project. Self-ensembling for visual domain adaptation

Vyacheslav Shults, B17-DS-1
v.shults@innopolis.ru

November 2019

Abstract

This paper explores the use of self-ensembling for visual domain adaptation problems. This techniques is mostly based on this work [3] and is derived from the mean teacher variant [11] of temporal ensembling [8], a technique that achieved state of the art results in the area of semi-supervised learning. I introduce several modifications to the current approach for increasing performance of the domain adaptation. These modifications keep state of the art result as in the previous [3] work for SVHN → MNIST task. Moreover, this algorithm achieves close to the classifier trained in a supervised fashion.

1 Introduction

The strong performance of deep learning in computer vision tasks comes at the cost of requiring large datasets with corresponding ground truth labels for training. Such datasets are often expensive to produce, owing to the cost of the human labour required to produce the ground truth labels.

Semi-supervised learning is an active area of research that aims to reduce the quantity of ground truth labels required for training. It is aimed at common practical scenarios in which only a small subset of a large dataset has corresponding ground truth labels. Unsupervised domain adaptation is a closely related problem in which one attempts to transfer knowledge gained from a labeled source dataset to a distinct unlabeled target dataset, within the constraint that the objective (e.g.digit classification) must remain the same. Domain adaptation offers the potential to train a model using labeled synthetic data – that is often abundantly available – and unlabeled real data.

Recent work [11] has demonstrated the effectiveness of self-ensembling with random image augmentations to achieve state of the art performance in semisupervised learning benchmarks.

Authors of the original paper [3] achieved excellent result and I could reproduce it in my experiments but when I visualized the latent space of the

embedded vectors I saw that the distributions of feature-vectors of the classes has high variance as it shown in fig. 1.

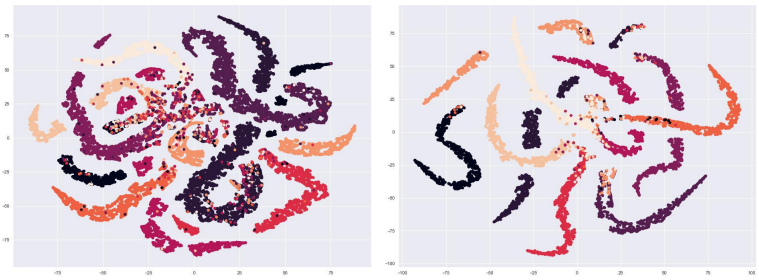

Figure 1: t-SNE visualization of features extracted from SVHN-test (left) and MNIST-test (right) by encoder trained using original method [3] on SVHN-train. It can be seen that the projected distributions are elongated and convoluted.

After discovering this effect I set my goal to straighten the distributions to make them look like a normal distribution, preserving or improving the current result. From face recognition task it is a well-known problem to straighten feature vector space because in datasets there are dozens of thousands of distinct people faces and distinguishing between them is the key problem that is had to to be solved. According to the overview paper [12] the most recent and the best method for straighten of the space is the Arc Face [2] module. Even though this module straightens the space, learning using it alone will degrade the result (up to 0.8 accuracy from the 0.99). Therefore additional modifications are needed.

2 Method

Baseline method builds upon the mean teacher semi-supervised learning model [11] which is described in the [3]. My improvement to the algorithm is that an Arc Face module was added as additional classifier to the existing one. Then the logits given by Arc Face module and Linear classifier are concatenated and goes into the Cross-entropy loss. Impact of the logits from Arc Face module make latent space of the features more straightened meanwhile Linear classifier tries to separate this space.

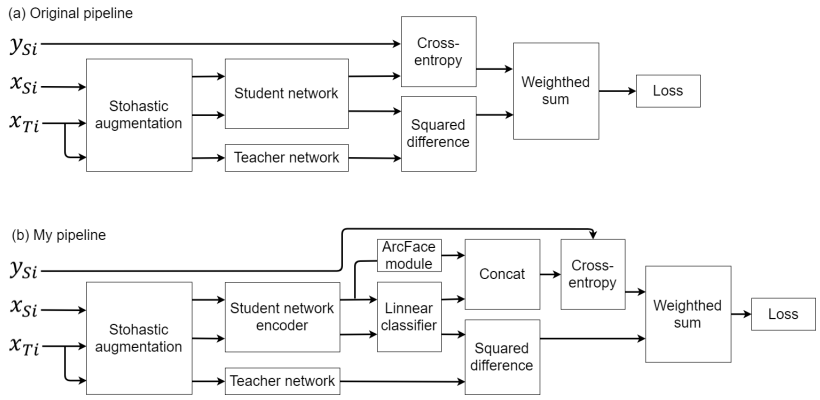

Figure 2: The network structures of the original training pipeline and my pipeline. In my solution labels of the source data also duplicated to be compatible with the dimensionality of the result of concatenation.

3 Experiments

Implementation of the original paper was develop using PyTorch and is publically available at http://github.com/Britefury/self-ensemble-visual-domain-adapt. My implementation also develop using PyTorch and use several modules and codelines from original paper.

3.1 Training setup

For my experiments I firstly used modified ResNet-18 [5] where instead of standard residual block inverted residual with Squeeze and Excitation blocks [6] from EfficientNet architecture [10]. This model due to its small number of parameters achieved only 0.93 accuracy. After that I start using architecture from paper [3] to compare experiments can be easier. I used Adam optimizer [7] with One Cycle Scheduler [9] for increasing speed of convergence. I used the same augmentations and parameters of the optimizer as in the original paper [3].

3.2 Dataset

Google's SVHN (Street View House Numbers) is a colour digits dataset of house number plates. This approach significantly outpaces other techniques and achieves an accuracy close to that of supervised learning.

3.3 Results

Method class	Method name	Accuracy
Train on Sourse	Effnet ArcFace	74.0
Train on Sourse	SupSrc+TFA[3]	71.73
Train on Target	SupTgt+TF[3]	99.61
GAN	G2A [1]	84.70
GAN	ADA [4]	97.6
Mean Teacher	MT+CT+TFA [3]	99.26
Mean Teacher	My result	99.19

Table 1: Benchmark classification accuracy

Method name	Src Train	Src Test	Trg Test
Source Train Effnet ArcFace Swish	94.0	93.8	73.6
MT InvResnet ArcFace+Linear Swish	94.0	95.4	92.9
MT SimpleNet ArcFace Relu	93.7	95.1	80.1
MT SimpleNet ArcFace+Linear Swish	92.5	94.8	99.19
MT SimpleNet ArcFace+Linear ReLu	95.3	95.86	99.29

Table 2: Benchmark classification accuracy among the experiments conducted in the course of the study. MT holds for Mean Teacher, algorithms presented in Section 2. ArcFace+Linear means that the output of the encoder is input for both Linear classification layer and Arc Face model. SimpleNet is the network presented in [3] as a model for SVHN $\leftrightarrow$ MNIST task.

3.4 Visualization

In this section I present visualization of the latent space that was projected using t-SNE algorithm. After application of my method the distributions of classes became more compact. Also distance between them increased.

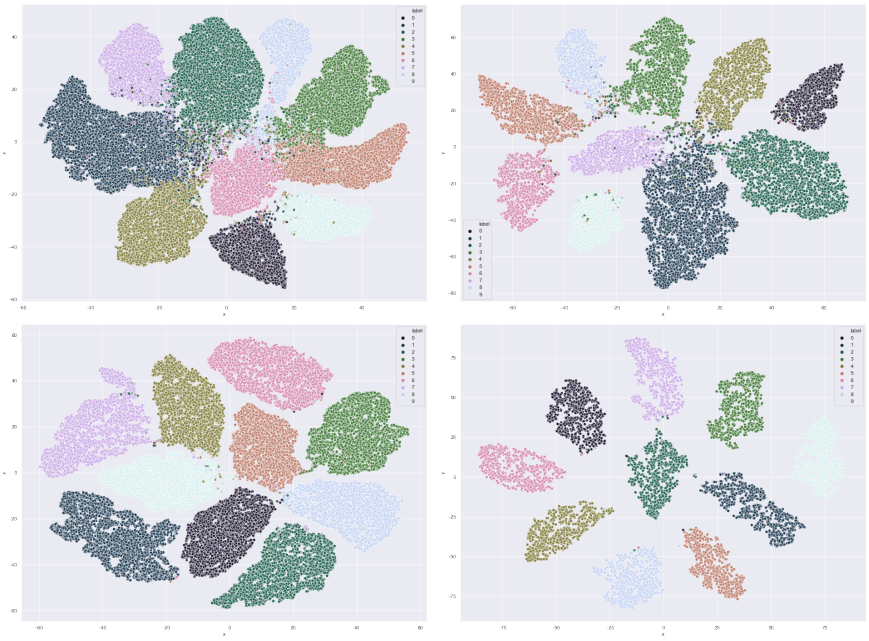

Figure 3: Visualization of SVHN-train (left upper), SVHN-test (right upper), MNIST-train (left bottom), MNIST-test (right bottom) given from MT SimpleNet ArcFace+Linear Swish model.

Also it is interesting to compare latent spaces before domain adaptation and after. There is no need to compare with latent space before training because there will be noise features that bring nothing except mess. For model before DA I used Source Train Effnet ArcFace Swish and for model after DA - MT SimpleNet ArcFace+Linear Swish.

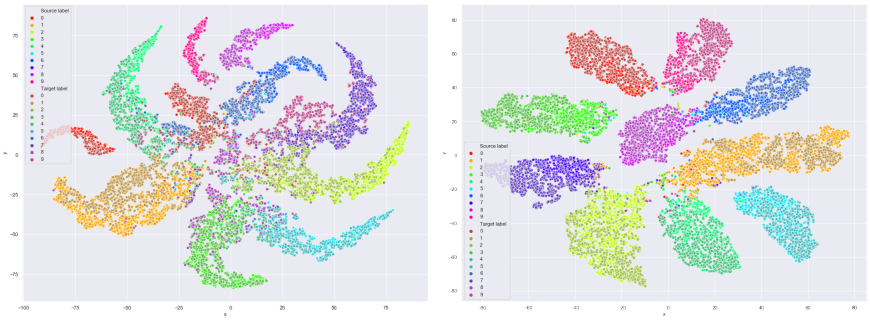

Figure 4: Visualization of feature projection of SVHN-test and MNIST-test before domain adaptation (left) and after (right). It is noticeable that after DA distribution of source and target features almost equal. For before DA case some target class distributions are close to the source class distributions but another like 0's or 9's are not.

4 Conclusion

In conclusions I can say that I solved domain adaptation problem for SVHN → MNIST task completely. I learned how to apply mean teacher algorithm to DA tasks and in future try to do solve another problem for another datasets with this method. I'm glad that my hypothesis about rectification of latent space with ArcFace confirmed. I think that there is nothing to add to increase performance because it is already close to supervised score.

References

[1] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks, 2016.

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition, 2018.

[3] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation, 2017.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2017.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[8] Laine S. and Aila T. Temporal ensembling for semi-supervised learning, 2017.

[9] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2017.

[10] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2019.

[11] Valpola H. Tarvainen A. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2017.

[12] Mei Wang and Weihong Deng. Deep face recognition: A survey, 2018.