ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Adversarial domain adaptation *without* gradient reversal layer

Hugo Serieys and Aymen Cherif [*]

EURA NOVA - Research and Development
Mont-Saint-Guibert - Belgium

Abstract. Adversarial domain adaptation is one of the most efficient way to deal with the domain shift phenomenon that domain adaptation tries to solve. We propose an improvement to the popular GRL method introduced in [9], an unsupervised domain adaptation (i.e. no labels in the target domain) technique easy to implement. We call our method NoGRL, and it is inspired from generative adversarial networks [11]. Our main idea is to dissociate prediction optimization and domain adaptation optimization. Our method outperforms results obtained by GRL in small images benchmarks.

1 Introduction

Deep learning has become a very powerful and popular technology in the past few years, especially on computer vision tasks with convolutional neural networks. To achieve high performances, deep learning models require huge amounts of labeled data to be trained properly. This is the main drawback of this technology. If gathering large amounts of data is nowadays easier than ever, labeling this huge quantity of such data is complex and expensive as it has to be done manually. Moreover, another well-known problem of deep learning models is the *domain shift* phenomenon: the ability of a model to generalize well from a training dataset to new ones is limited.

To deal with these two big issues of deep learning, a recent research field has appeared: domain adaptation. The goal of this research topic is to leverage already available labeled data from a given domain, called *source domain*, to train a model in a specific way to be efficient on target data from a different but related domain to the first one, called *target domain*.

As a matter of fact, domain adaptation were already studied on classical machine learning algorithms such as support vector machines. Nevertheless, efficiency has become way better when doing domain adaptation with deep learning models.

Our method presented in this paper aims to improve an already existing and popular domain adaptation called GRL introduced in [9]. Our paper is organised as follows; in Section 2 we will present and discuss related work to give a research context and the basis of our method; the method introduced in this paper is developed in Section 3 and experiments plus results obtained from it are presented in Section 4; finally the conclusion will be in Section 5.

[*]The elaboration of this scientific paper was supported by the Ministry of Economy, Industry,Research, Innovation, IT, Employment and Education of the Region of Wallonia (Belgium),through the funding of the industrial research project Jericho (convention no. 7717).

2 Related work

2.1 Unsupervised domain adaptation

Unsupervised domain adaptation is when labels in the target domain are not available. Different deep domain adaptation methods exist to deal with this situation, the first ones inspired from shallow discrepancy-based methods. MMD has been used in many methods, providing good results on standard datasets, such as in [1] where they introduce a MMD variant *Joint Maximum Mean Discrepancy*. [2] also uses another variant of MMD with an architecture in two parts where the first part is shared for source and target domains, here to extract general features and is then divided in two parts to get each domain specificity, and MK-MMD is introduced to minimize shift between each part. Another approach adapted from shallow methods is Deep CORAL [3]. This method introduces a specific loss called CORAL close to others cited above and with advantage to be easy to add in a regular deep neural network. Finally, the most efficient methods for deep domain adaptation are based on adversarial training as we will see now.

2.2 Adversarial domain adaptation

Most of the recent research on domain adaptation use adversarial methods. Different methods directly use architecture inspired from GAN ([11]) as in [4] or in [5]. In [5], they use multiple steps training including a step where a kind of GAN is trained: instead of real and fake inputs, they are replaced by source and target inputs. This method provides quite promising results even if it is hard to optimize (three training steps imply three times more hyperparameters...).

One of the most popular adversarial domain adaptation approach, detailed in [9], is a simple yet efficient technique that can be added in any regular neural network consisting of a domain classifier optimized with a gradient reversal layer (GRL) that enables adversarial training with only one loss. This eases the implementation and is totally transparent during training. This method has been reused multiple times in more complex architectures such as in [6], [7] or [8]. This last reference leverage gradient reversal layer inside a very complex architecture called TADA using attention mechanisms to help the model focusing on the most transferable parts of input images from source to target domain.

3 No gradient reversal layer (NoGRL)

3.1 Unsupervised domain adaptation

The goal of this paper is to propose an algorithm to improve unsupervised domain adaptation on classification task. This problem consists in enabling generalization capability of a classifier across domains with unlabeled data from target domain. In a more formal way, this means having two domains following different probability distributions: a labeled source domain $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and an unlabeled target domain $\mathcal{D}_t = \{x_i^t\}_{j=1}^{n_t}$ where x_i is a sample and y_i the label associated ; n_s and n_t respectively being the number of source and target samples.

3.2 Gradient reversal layer approach

We follow the approach presented in [9]. The architecture of the neural network used is composed of three different parts. The feature extractor, $G_f(\cdot;\theta_f)$, with parameters θ_f, is composed of multiple convolutional layers and is trained to extract domain invariant features from input samples. The classifier, $G_y(\cdot;\theta_y)$, with parameters θ_y, is composed of fully connected layers and is trained to classify input into different known categories. Finally, the domain classifier, $G_d(\cdot;\theta_d)$, with parameters θ_d, is composed of fully connected layers and is trained to classify extracted features from G_f between source or target domain.

The mechanism used for actual domain adaptation is to train G_f and G_d in an adversarial way: the objective of G_d is to be as efficient as possible in domain classification and the objective of G_f is to extract features from input samples in such a way that G_d can not distinguish whether it is from source or target domain.

To achieve such a behavior, [9] introduces two losses, the prediction loss $\mathcal{L}_y$ and the domain loss $\mathcal{L}_d$. The prediction loss is used to train the model to be efficient for the classification task, meanwhile the domain loss is used to train the model for a proper domain adaptation.

The trick used by [9] to train G_f and G_d in an adversarial way with only one loss regarding domain adaptation is done during backpropagation by computing a flip on the gradient from the [last G_d layer (i.e. the one just before G_f)] and changes its sign, i.e., multiplies it by -1, before passing it to the preceding layer [9]. This mechanism, called GRL (Gradient Reversal Layer) provides good results and has the big advantage to be easy to implement with existing deep learning frameworks.

3.3 Replacing gradient reversal layer

What we introduce in this paper is another way to implement such a mechanism that we call NoGRL, inspired by Generative Adversarial Networks, introduced in [11], by adding another loss $\mathcal{L}_c$ that we call the confusion loss. This modification doesn't change anything for the classification task and for the domain classification task regarding G_d. The change is for G_f training regarding domain adaptation: instead of being updated thanks to the GRL mechanism, G_f is updated thanks to the new confusion loss that is basically the domain loss applied with fake domain labels.

The following loss definitions are done on the i-th example, where $\mathcal{L}^i$ is the loss of the i-th example, and then the global loss can be defined as:

$$\mathcal{L}(\{(x_i, y_i) \in \mathcal{D}\}) = \frac{1}{n} \sum_{x_i \in \mathcal{D}} \mathcal{L}^i(x_i; y_i). \quad (1)$$

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

The prediction loss $\mathcal{L}_y$ chosen is the popular cross-entropy loss and is only applied on samples from source as we do unsupervised domain adaptation. The domain loss $\mathcal{L}_d$ chosen is the binary cross-entropy, applied on domain classifier only.

Finally, unlike [9], we introduce a third loss, that we call the confusion loss, applied on feature extractor only, as defined by

$$\begin{gathered}\forall (x_i, y_i) \in D_s \cup D_t, \\ \mathcal{L}_c^i(G_d(G_f(x_i, \theta_f), \theta_d), d_i) = \\ (1 - d_i)H(G_d(G_f(x_i, \theta_f), \theta_d)) + d_i H(1 - G_d(G_f(x_i, \theta_f), \theta_d))\end{gathered} \tag{2}$$

where H is the cross-entropy loss function, and $d_i \in 0, 1$ denotes the domain label (source or target) for the i-th example, which indicates whether x_i comes from the source or the target domain. $\mathcal{L}_c$ is very similar to $\mathcal{L}_d$ as $\mathcal{L}_c$ **actually means using $\mathcal{L}_d$ with fake domain labels**.

The training objective is finally to optimize the following function:

$$E(\theta_f, \theta_y, \theta_d) = \alpha \mathcal{L}_y(\theta_f, \theta_y) + (1 - \alpha)(\mathcal{L}_d(\theta_f, \theta_d) + \mathcal{L}_c(\theta_f, \theta_d)). \tag{3}$$

where alpha, inspired by λ parameter in [9], is defined by $\alpha = \frac{1}{1+e^{-\gamma p}}$ with γ is an hyperparameter and p is the training progress linearly changing from 0 to 1.

The function to optimize in (3) has been built in order to differentiate prediction training and domain adaptation training. Indeed, the α parameter enables to put the training focus on domain adaptation task by emphasizing $\mathcal{L}_d$ and $\mathcal{L}_c$ first and then on prediction task by emphasizing $\mathcal{L}_y$. $\mathcal{L}_d$ and $\mathcal{L}_c$ are optimized simultaneously to avoid unbalanced training between feature extractor and domain classifier, which is mandatory as they are trained in an adversarial way.

4 Experiments

We now evaluate NoGRL on an image classification task in an unsupervised way. This is done on one popular benchmark: images of digits. Digits datasets include MNIST, USPS and SVHN representing handwritten digits and house number digits.

Architecture used for experiments is based on [9]. The only significant difference is the feature extractor: we used a VGG-16 [10] network pre-trained on ImageNet from Keras Applications[1]. We also added dropout layers on classifier and domain classifier after each fully-connected layers (rate = 0.5) and L2 regularization on each fully-connected layer (value = 0.01).

We set the α function parameter $\gamma = 10$ which gave the best results. We used stochastic gradient descent (SGD) optimizer with a momentum term of 0.9, as in [9].

Results on digits datasets can be seen in Table 1. *Source only* column consists of the results from a model trained using supervised training with source domain

[1]See Keras Application Websitehttps://keras.io/applications/#vgg16

Table 1: Results on digits datasets

Source	Target	Source only	Domain adaptation method			Target fully supervised
		VGG-16	GRL		NoGRL	
			Paper	Ours		
MNIST	USPS	75.2±1.6	77.1±1.8	82	95.3	98.9±0.1
USPS	MNIST	57.1±1.7	73.0±2.0	72.2	97.2	99.2±0.1
SVHN	MNIST	60.1±1.1	73.9±1.8	76.3	84.7	99.2±0.1

only, and evaluated on target domain: it represents the expected baseline. *Target fully supervised* column consists of results from a model trained using supervised training with target domain and evaluate on it as well: it represents the best achievable result. We evaluate our methods on three common configurations for these datasets: MNIST⇒USPS ; USPS⇒MNIST and SVHN⇒MNIST.

MNIST⇒USPS and USPS⇒MNIST: These configurations have the smallest domain shift. MNIST and USPS both consist of grayscale images of handwritten digits. We set both datasets images to 28x28 dimensions: as is for MNIST, and USPS images were enlarged using nearest-neighbor interpolation. As one can see, NoGRL outperforms GRL [9] results, enhancing the accuracy by **at least 13%** (compared to our results with GRL). Results are very close to the best achievable results: less than 4% difference in accuracy.

SVHN⇒MNIST: This configuration is the most challenging for the digits. Domain shift between SVHN and MNIST is bigger than before: SVHN images contain colors and can contain two digits - misleading the classification - whereas MNIST are grayscale images of handwritten digits. We set both datasets images to 32x32 dimensions: as is for SVHN, and MNIST images were enlarged using nearest-neighbor interpolation and formatted into RGB by duplicating pixels value for each RGB layer. As one can see, NoGRL again outperforms GRL [9] results, enhancing the accuracy by **more than 8%**. Unlike other configurations described above, results are still far from the best achievable results but it is still a promising improvement.

5 Conclusion

This paper presented NoGRL, an improvement to the existing and popular GRL method [9] used in domain adaptation. The main motivation behind our contribution is to dissociate prediction and domain adaptation during training: domain adaptation is emphasized during the first part of the training, and then the focus is put on prediction optimization. As a result, the method introduced, simple yet efficient, improves results obtained by the original GRL method [9]. Such results are promising as many recent domain adaptation papers (e.g. [7] or [8]) are using GRL in their pipeline and our method could help getting even better performances in domain adaptation research by simply adapting these existing models by including NoGRL.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Future work could then be to implement NoGRL on the latest domain adaptation methods currently using GRL to measure how beneficial our approach can be in comparison to GRL [9].

References

[1] Mingsheng Long, Han Zhu, Jianmin Wang and Michael I. Jordan. Deep Transfer Learning with Joint Adaptation Networks, 2016; arXiv:1605.06636.

[2] Mingsheng Long, Yue Cao, Jianmin Wang and Michael I. Jordan. Learning Transferable Features with Deep Adaptation Networks, 2015; arXiv:1502.02791.

[3] Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation, 2016; arXiv:1607.01719.

[4] Swami Sankaranarayanan, Yogesh Balaji, Carlos D. Castillo and Rama Chellappa. Generate To Adapt: Aligning Domains using Generative Adversarial Networks, 2017; arXiv:1704.01705.

[5] Han Zou, Yuxun Zhou, Jianfei Yang, Huihan Liu, Hari Prasanna Das and Costas J Spanos. Consensus adversarial domain adaptation, 2019.

[6] Pedro O. Pinheiro. Unsupervised Domain Adaptation with Similarity Learning, 2017; arXiv:1711.08995.

[7] You Kaichao, Long Mingsheng, Cao Zhangjie, Wang Jianmin and Jordan Michael I. Universal domain adaptation, 2019. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[8] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation, 2019. The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), February 2019.

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, FranÃ§ois Laviolette, Mario Marchand and Victor Lempitsky. Domain-Adversarial Training of Neural Networks, 2015, Journal of Machine Learning Research 2016, vol. 17, p. 1-35; arXiv:1505.07818.

[10] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014; arXiv:1409.1556.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio. Generative Adversarial Networks, 2014; arXiv:1406.2661.