Domain Adaptation

Domain Adaptation [10]

- Domain Adaptation is a field associated with machine learning and transfer learning.
- **Learning a discriminative classifier or other predictor in the presence of a shift between training and test distributions is known as domain adaptation (DA).**
- This scenario arises when we aim at learning from a source data distribution a well performing model on a different (but related) target data distribution.
- **For instance, one of the tasks of the common spam filtering problem consists in adapting a model from one user (the source distribution) to a new one who receives significantly different emails (the target distribution).**
- Note that, when more than one source distribution is available the problem is referred to as multi-source domain adaptation.

Types of Domain Adaptation [10]

- There are several contexts of domain adaptation. They differ in the information considered for the target task.

  - **The unsupervised domain adaptation:** the learning sample contains a set of labelled source examples, a set of unlabelled source examples and an unlabelled set of target examples.

  - **The semi-supervised domain adaptation:** in this situation, we also consider a "small" set of labelled target examples.

  - **The supervised domain adaptation:** all the examples considered are supposed to be labelled.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Learning a discriminative classifier or other predictor in the presence of a shift between training and test distributions is known as domain adaptation (DA).

Figure 2. Examples of domain pairs used in the experiments. See Section 4.1 for details.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

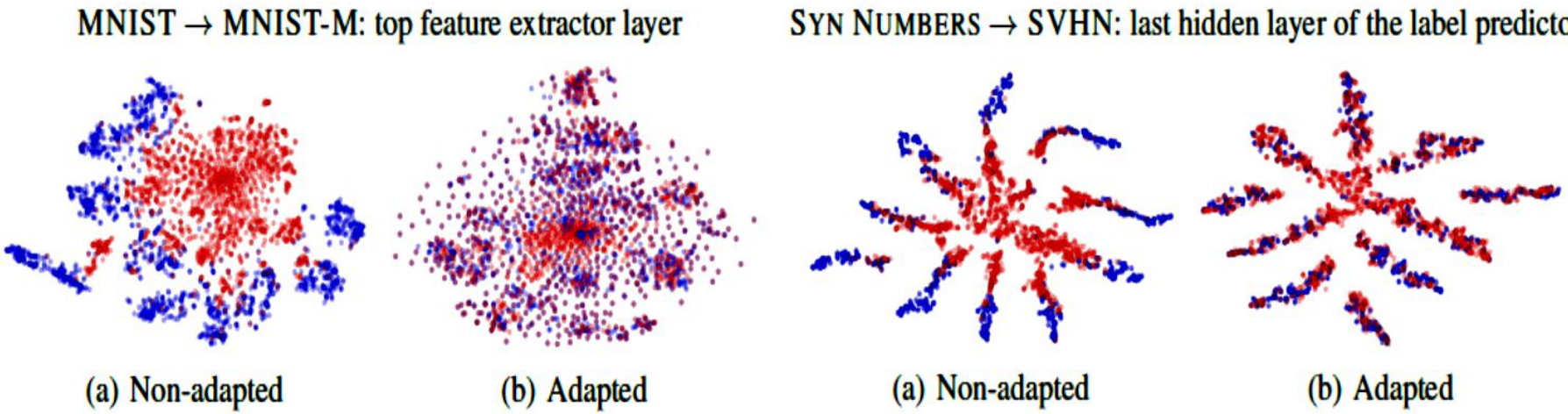

Figure 3. The effect of adaptation on the distribution of the extracted features (best viewed in color). The figure shows t-SNE (van der Maaten, 2013) visualizations of the CNN's activations (**a**) in case when no adaptation was performed and (**b**) in case when our adaptation procedure was incorporated into training. *Blue* points correspond to the source domain examples, while *red* ones correspond to the target domain. In all cases, the adaptation in our method makes the two distributions of features much closer.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Problem Statement
  - Given a labelled source domain and an unlabelled target domain, we would like to train a classifier or a predictor which would give accurate predictions on the target domain.

- Assumptions
  - Probability distribution of source domain is not equal to the probability distribution of target domain.

  - Source dataset is labelled. Target dataset is unlabelled.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Goal
  - Perform some transformation on the source and target domain and bring the transformed distributions closer. Then, Train the classifier on the transformed source distribution and since both the transformed distributions are now similar, The model will achieve better accuracy on the target domain during test time.

  - In-order to perform the transformation, a neural network is used.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

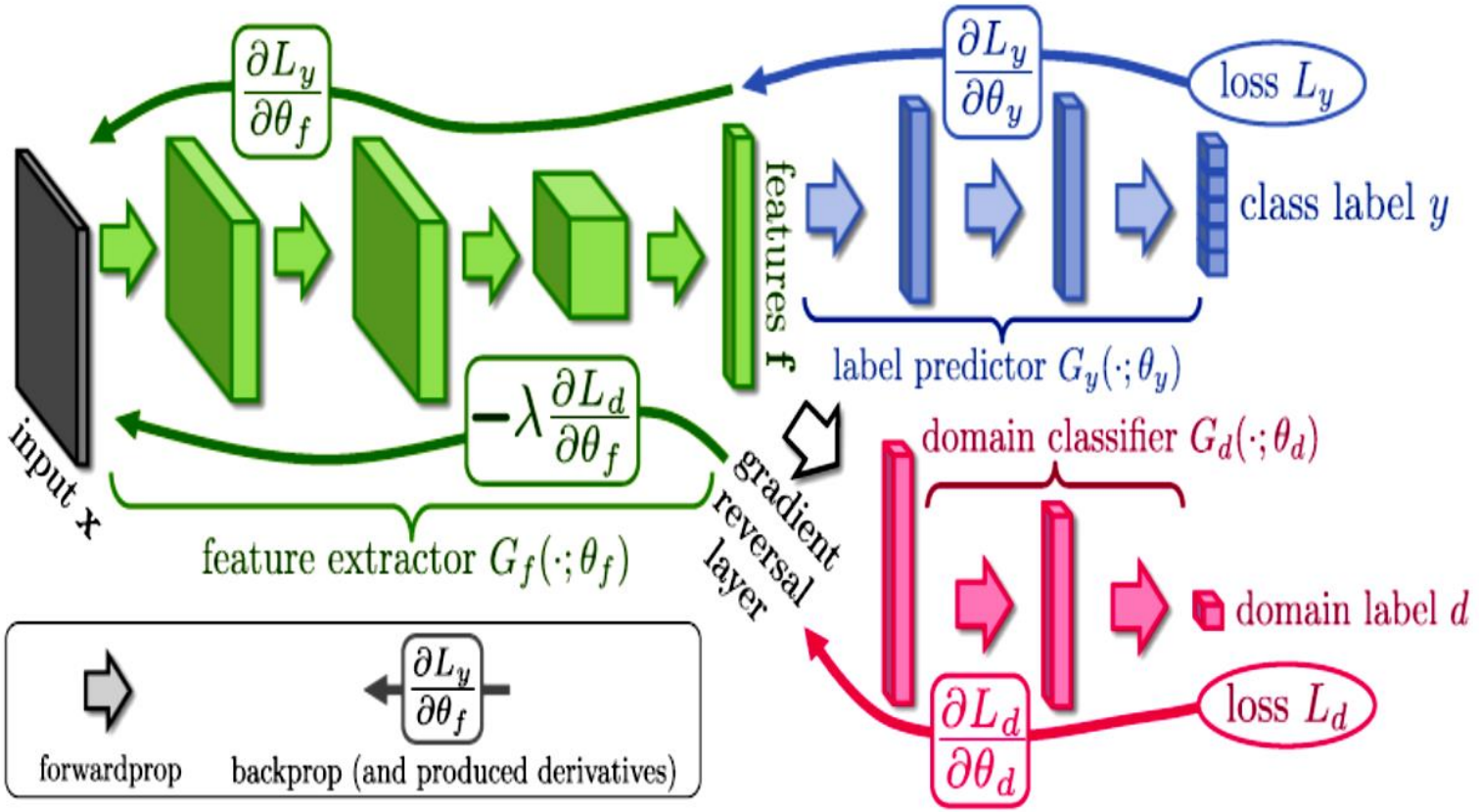

Figure 1. The **proposed architecture** includes a deep *feature extractor* (green) and a deep *label predictor* (blue), which together form a standard feed-forward architecture. Unsupervised domain adaptation is achieved by adding a *domain classifier* (red) connected to the feature extractor via a *gradient reversal layer* that multiplies the gradient by a certain negative constant during the backpropagation-based training. Otherwise, the training proceeds in a standard way and minimizes the label prediction loss (for source examples) and the domain classification loss (for all samples). Gradient reversal ensures that the feature distributions over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus resulting in the domain-invariant features.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- **Feature Extractor:**
  - This is a neural network that will learn to perform the transformation on the source and target distribution.

- **Label Classifier:**
  - This is a neural network that will learn to perform classification on the transformed source distribution. Since, source domain is labelled.

- **Domain Classifier:**
  - This is a neural network that will be predicting whether the output of the Feature Extractor is from the source or the target distribution.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- **Idea:**
  - At training time, in order to obtain domain-invariant features, we seek the parameters θ_f of the feature mapping that maximize the loss of the domain classifier (by making the two feature distributions as similar as possible (mind the word feature distribution and not source distribution or target distribution)), while simultaneously seeking the parameters θ_d of the domain classifier that minimize the loss of the domain classifier.

  - In addition, we seek (the parameters θ_y) to minimize the loss of the label predictor.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- **Intuition:**
  - Basically, the feature extractor will try to perform some transformation on the source and target instances such that the transformed instances appear as if it is coming from the same distribution and domain classifier will not be able to classify the domain of the transformed instances.
  - This is achieved by training both the feature extractor and domain classifier in such a way that feature extractor will be trained to maximize the domain classification loss, while domain classifier will try to minimize the domain classification loss.
  - So, this is similar to adversarial training wherein the feature extractor is trying to confuse domain classifier by bringing the two distributions closer.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- **Intuition:**
  - For the transformed source instances the label predictor will be trained for predicting the labels of the source instances.
  - The feature extractor will therefore be trained to minimize classification loss of the label predictor and maximize classification loss of domain predictor.
  - The label predictor and domain predictor will be trained to minimize there respective classification loss.
  - Thus using the above three components, The Feature extractor will learn to produce discriminative and domain-invariant features.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Optimization with Backpropagation

$$E(\theta_f, \theta_y, \theta_d) = \sum_{\substack{i=1..N \\ d_i=0}} L_y\left(G_y(G_f(\mathbf{x}_i; \theta_f); \theta_y), y_i\right) -$$

$$\lambda \sum_{i=1..N} L_d\left(G_d(G_f(\mathbf{x}_i; \theta_f); \theta_d), y_i\right) =$$

$$= \sum_{\substack{i=1..N \\ d_i=0}} L_y^i(\theta_f, \theta_y) - \lambda \sum_{i=1..N} L_d^i(\theta_f, \theta_d) \qquad (1)$$

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Optimization with Backpropagation

$$\theta_f \longleftarrow \theta_f - \mu\left(\frac{\partial L_y^i}{\partial \theta_f} - \lambda\frac{\partial L_d^i}{\partial \theta_f}\right) \tag{4}$$

$$\theta_y \longleftarrow \theta_y - \mu\frac{\partial L_y^i}{\partial \theta_y} \tag{5}$$

$$\theta_d \longleftarrow \theta_d - \mu\frac{\partial L_d^i}{\partial \theta_d} \tag{6}$$

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Gradient Reversal Layer
  - For training the feature extractor in order to maximize the classification loss of domain predictor, Gradient Reversal layer was place between Feature extractor and domain classifier.
  - The Gradient Reversal Layer basically acts as an identity function (outputs is same as input) during forward propagation but during back propagation it multiplies its input by $-\lambda$.
  - Intuitively, During back propagation the output of GRL is basically leading to the opposite of Gradient descent that is performing gradient ascent on the feature extractor with respect to the classification loss of Domain predictor.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Results

Method	Source	MNIST	Syn Numbers	SVHN	Syn Signs
	Target	MNIST-M	SVHN	MNIST	GTSRB
Source only		.5749	.8665	.5919	.7400
SA (Fernando et al., 2013)		.6078 (7.9%)	.8672 (1.3%)	.6157 (5.9%)	.7635 (9.1%)
Proposed approach		.8149 (57.9%)	.9048 (66.1%)	.7107 (29.3%)	.8866 (56.7%)
Train on target		.9891	.9244	.9951	.9987

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Results

METHOD (SOURCE / TARGET)	AMAZON / WEBCAM	DSLR / WEBCAM	WEBCAM / DSLR
GFK(PLS, PCA) (GONG ET AL., 2012)	.464 ± .005	.613 ± .004	.663 ± .004
SA (FERNANDO ET AL., 2013)	.450	.648	.699
DA-NBNN (TOMMASI & CAPUTO, 2013)	.528 ± .037	.766 ± .017	.762 ± .025
DLID (S. CHOPRA & GOPALAN, 2013)	.519	.782	.899
DECAF$_6$ SOURCE ONLY (DONAHUE ET AL., 2014)	.522 ± .017	.915 ± .015	–
DANN (GHIFARY ET AL., 2014)	.536 ± .002	.712 ± .000	.835 ± .000
DDC (TZENG ET AL., 2014)	.594 ± .008	.925 ± .003	.917 ± .008
PROPOSED APPROACH	**.673 ± .017**	**.940 ± .008**	**.937 ± .010**

Accuracy evaluation of different DA approaches on the standard OFFICE (Saenko et al., 2010) dataset. Prposed method (last row) outperforms competitors setting the new state-of-the-art.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Results

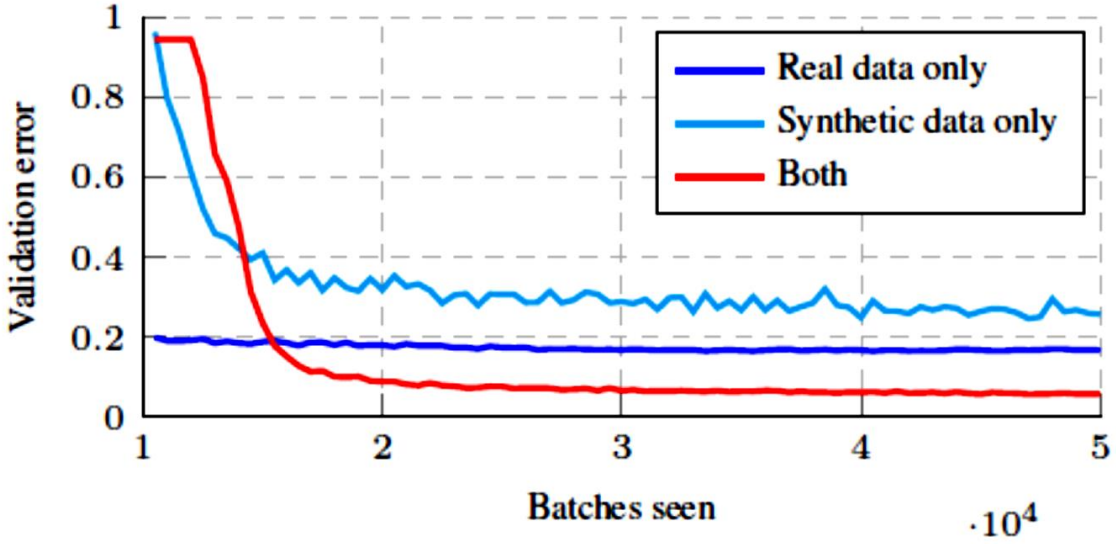

Figure 4. Semi-supervised domain adaptation for the traffic signs. As labeled target domain data are shown to the method, it achieves significantly lower error than the model trained on target domain data only or on source domain data only.

As an additional experiment, they also evaluated the proposed algorithm for semi-supervised domain adaptation, i.e. when one is additionally provided with a small amount of labeled target data. For that purpose we split GTSRB into the train set (1280 random samples with labels, original size > 50K images) and the validation set (the rest of the dataset). The validation part is used solely for the evaluation and does not participate in the adaptation. The training procedure changes slightly as the label predictor is now exposed to the target data.

Unsupervised Domain Adaptation by Backpropagation [11, 12]

- Pros
  - Brings both domains closer after performing some transformation
  - Learns discriminative and domain invariant features

- Cons
  - Gradient Reversal Layers leads to vanishing gradient problem once the domain predictor has achieved good accuracy

References

1. www.ntu.edu.sg/home/sinnopan/.../A%20Survey%20on%20Transfer%20Learning.ppt

2. http://ruder.io/transfer-learning/

3. http://imatge-upc.github.io/telecombcn-2016-dlcv/slides/D2L5-transfer.pdf

4. Olivas, Emilio Soria, ed. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques. IGI Global, 2009.

5. http://cs231n.github.io/transfer-learning/

References

6. https://machinelearningmastery.com/transfer-learning-for-deep-learning/

7. Rusu, A. A., Vecerik, M., Rothörl, T., Heess, N., Pascanu, R., & Hadsell, R. (2016). Sim-to-Real Robot Learning from Pixels with Progressive Nets. arXiv Preprint arXiv:1610.04286.

8. Johnson, M., Schuster, M., Le, Q. V, Krikun, M., Wu, Y., Chen, Z., … Dean, J. (2016). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation.

9. Sun, B., Feng, J., & Saenko, K. (2016). Return of Frustratingly Easy Domain Adaptation. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

References

10. https://en.wikipedia.org/wiki/Domain_adaptation

11. https://pdfs.semanticscholar.org/presentation/f4af/de757b9dfc697d149e95cb193aa4749530e2.pdf

12. Ganin, Yaroslav, and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation." arXiv preprint arXiv:1409.7495 (2014).

13. https://medium.com/@2017csm1006/unsupervised-domain-adaptation-by-backpropagation-da730a190fd2

Disclaimer

- These slides are not original and have been prepared from various sources for teaching purpose.