RDFIA - Domain Adaptation (2-c)

Bourzag Mohamed Chakib, Missoum Youcef

December 2025

1 Introduction

This document contains the written solutions for the **domain adaptation** practical work (RDFIA TP2-c). Let's start with a summary of this subject before moving to the questions.

2 Summary of the Practical

2.1 Context and Motivation

In standard supervised learning, we assume that the training data (source domain) and the test data (target domain) are drawn from the same probability distribution. However, in real-world applications, this assumption often fails due to **domain shift**. For instance, a model trained on black-and-white handwritten digits (MNIST) may fail when applied to colored digits with complex backgrounds (MNIST-M), despite the semantic content (the numbers) remaining the same.

The goal of **Unsupervised Domain Adaptation (UDA)** is to transfer the knowledge learned from a labeled source domain $\mathcal{D}_S$ to an unlabeled target domain $\mathcal{D}_T$, bridging the gap between their distributions. In this practical, we address this challenge using **Domain-Adversarial Neural Networks (DANN)**, a method that learns features which are discriminative for the classification task but invariant with respect to the domain shift.

2.2 Theoretical Framework: DANN and GRL

The DANN architecture proposes a game-theoretic approach similar to GANs, consisting of three components:

- **Feature Extractor (G_f, Green):** Maps input images to a latent feature vector.
- **Label Predictor (G_y, Blue):** Classifies the digit based on the latent features.
- **Domain Classifier (G_d, Pink):** Attempts to predict the domain (Source vs. Target) from the latent features.

The core innovation is the **Gradient Reversal Layer (GRL)**. During forward propagation, the GRL acts as an identity function. However, during backpropagation, it multiplies the gradient flowing from the domain classifier by a negative scalar $-\lambda$. This creates an adversarial objective:

$$E(\theta_f, \theta_y, \theta_d) = \mathcal{L}_y(\theta_f, \theta_y) - \lambda \mathcal{L}_d(\theta_f, \theta_d) \tag{1}$$

where we seek to minimize the classification error $\mathcal{L}_y$ while *maximizing* the domain classification error $\mathcal{L}_d$. This forces the Feature Extractor to construct representations that are **domain-agnostic**, thereby aligning the two distributions in the feature space.

2.3 Practical Objectives

In this work, we explored the impact of this adversarial alignment through several steps:

1. **Baseline Evaluation:** We first trained a model without domain adaptation (Source only). As expected, while it achieved high accuracy on the source test set, it performed poorly on the target domain due to the domain shift.
2. **Implementation of GRL:** We integrated the Gradient Reversal Layer to enable end-to-end adversarial training.
3. **Adversarial Training:** By training the full DANN architecture, we observed that the *domain loss* increased (indicating confusion), while the *target accuracy* significantly improved, validating the successful alignment of features.
4. **Feature Visualization:** Using techniques like t-SNE (implied context), we could visually verify that source and target features—previously separated into distinct clusters—became overlapping after adaptation.

3 Questions

1. If you keep the network with the three parts (green, blue, pink) but didn't use the GRL, what would happen ?

   Solution.
   The model would only learn to predict the right domain of each input. So, the feature representation would be quite different between source and target domains and the model would not be domain-agnostic. As a result, it can't be used to predict the target labels as it differentiates between source and target quite good.

   □

2. Why does the performance on the source dataset may degrade a bit ?

   Solution.
   Because the learnt feature space is a combination between the source and target domains. So, combining these two domains may lead to small sacrifices on the source domain to align to the target domain. The GRL forces the network to discard these specific source cues in favor of more generic ones. □

3. Discuss the influence of the value of the negative number used to reverse the gradient in the GRL.

   Solution.
   This negative number, denoted as $-\lambda$ (lambda), controls the trade-off between the classification objective and the domain alignment objective:

   - **If λ is too small (near 0):** The domain adversarial signal is too weak. The network will prioritize the Label Predictor, leading to good Source performance but poor domain alignment (features remain separated), resulting in poor Target performance.
   - **If λ is too large:** The domain confusion signal overpowers the classification signal. The Feature Extractor might destroy all useful semantic information just to confuse the domain classifier, causing the model to fail at the actual task (classifying digits) on both domains.
   - **Dynamic λ:** In practice (as in the original paper and the notebook), λ is often initialized at 0 and gradually increased during training to allow the Label Predictor to learn discriminative features first before enforcing alignment.

   □

4. Another common method in domain adaptation is pseudo-labeling. Investigate what it is and describe it in your own words.

   Solution.
   Pseudo-labeling is a semi-supervised learning technique used when we have, as in this case, labeled source data and unlabeled target data, but we define ground-truth labels on target data using the most confident model outputs after it hase been trained properly using the source data. More formally, we need to:

   (a) Train a model on the labeled Source data.
   (b) Use this model to predict labels for the unlabeled Target data.
   (c) Select the predictions with **high confidence** (e.g., probability > 0.9) and treat them as "**ground truth**" (pseudo-labels).
   (d) **Retrain (fine-tune)** the model on the combination of the Source data and the pseudo-labeled Target data.

   This iteratively pulls the decision boundary towards low-density regions of the Target distribution, effectively teaching the model using its own confident predictions on the new domain.

   □

5. **BONUS**:

   In the lab, trying to improve the results, by simply training it for 40 epochs instead of 20 epochs we notced a very good tradeoff between source and target:

Dataset	Epochs	Class Loss	Class Acc (%)	Domain Loss / Acc (%)
Source	20	0.09049	97.70	0.41486 / 94.71
	40	0.04266	98.70	0.60088 / 73.91
Target	20	1.81218	69.07	0.68872 / 44.49
	40	0.72701	**81.02**	0.63263 / 55.94

Table 1: Classification and domain adaptation performance on source and target datasets across training epochs.

We can notice a slight improvement in source performance for an even better one on target.

4 Conclusion

To conclude, this practical work successfully demonstrated the effectiveness of **Domain-Adversarial Neural Networks (DANN)** in mitigating the domain shift problem between standard MNIST and the more complex MNIST-M dataset.

By integrating a **Gradient Reversal Layer (GRL)**, we transformed a standard classifier into a domain-agnostic learner. Our experiments confirmed that without adaptation, the model relies heavily on domain-specific cues (like color or background texture), leading to poor generalization on the target domain. However, by optimizing the adversarial objective—simultaneously minimizing classification error and maximizing domain confusion—we forced the feature extractor to learn representations that are semantically rich but domain-invariant.

Our analysis highlighted several key insights:

- **The Trade-off:** Achieving domain invariance comes at a cost. As seen in our results, aligning the feature distributions requires discarding some source-specific information, which can lead to a marginal degradation in source accuracy (from nearly perfect to slightly lower), but this is a necessary sacrifice for substantial gains in target accuracy.
- **Stability and Convergence:** The adversarial minimax game is sensitive to hyperparameters, specifically the adaptation factor λ. However, as shown in our bonus experiment, extending the training to 40 epochs allowed the model to find a better equilibrium, significantly boosting target accuracy to **81.02%** while recovering source performance.
- **Alternative Approaches:** We also acknowledged that while adversarial alignment is powerful, other semi-supervised methods like *pseudo-labeling* offer viable alternatives by leveraging confident predictions to refine the decision boundary directly on the target domain.

Ultimately, this work illustrates that in the absence of labeled target data, "confusing" the model about the origin of the image is the key to achieving robust, generalized performance.