Deep Learning Practical Work 3-c
Domain Adaptation

Nicolas Thome   Guillaume Couairon   Mustafa Shukor   Asya Grechka
Matthieu Cord

Homework

- For all practical works, contact the supervisor of your group : `nicolas.thome@sorbonne-universite.fr`, `mustafa.shukor@isir.upmc.fr`, `guillaume.couairon@gmail.com` and `aagrechka @gmail.com`
- Email object must be `[RDFIA][Homework-3]`.
- After doing 3-d, you will need to finish by yourself at home the whole homework 3 (3-a, 3-b, 3-c, and 3-d). You will send by email this homework. You are encouraged to do it by group of two (please write both names in the email !)
  - You need to provide answers to all questions in one single PDF file. You can include inside the PDF code snippets (not a whole file !), graphs, or images that you deemed necessary.
  - Don't write long paragraphs of bullshit, we'll just get tired of it and be more severe.
  - Questions with a "Bonus" mark are optional. Questions with a ★ are worth double points.

Data, code, and PDF version of this file are available at
`https://rdfia.github.io`

Domain Adaptation

The input data can belong to different **domains**. A common example is the segmentation data used for autonomous driving. An image from the game GTA-5 and one taken in the street may represent the same classes (e.g. road, car, pedestrian, etc.) but the pixel distribution is widely different. Moreover, gathering labeled data can be very costly, both in time and in money. Therefore the field of domain adaptation aims to train on one labeled source dataset (like GTA-5 whose labels are given by the game engine) and to perform well on another unlabeled target dataset. See Fig. 1 for an actual example.

FIGURE 1 – Illustration of two datasets from two different domains : GTA-5 the game, and Cityscapes from Germand, Swiss, and French cities.

Figure 2 – Various pair of source/target datasets used in domain adaptation for image classification.

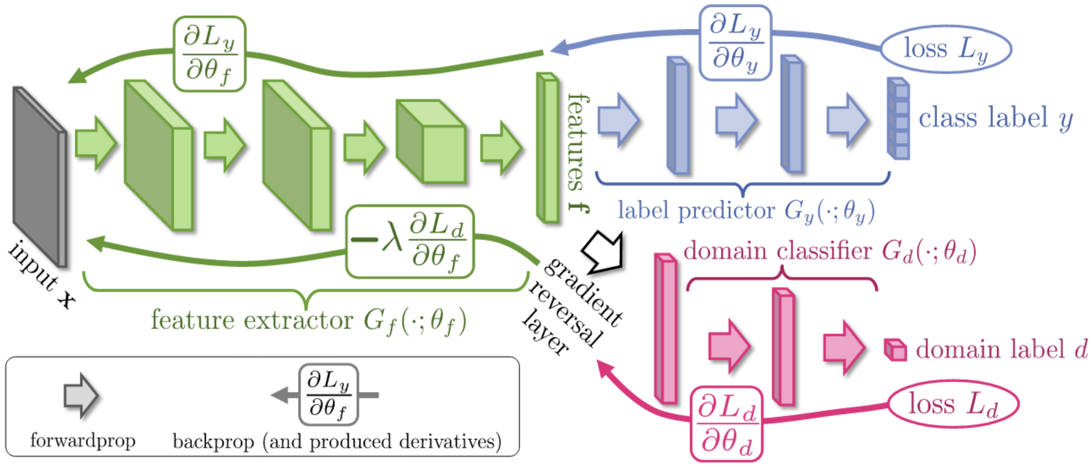

Figure 3 – DANN model with its GRL layer.

Partie 1 – DANN and the GRL layer

MNIST vs MNIST-M. In this session, we will build a model that will be trained on a labeled MNIST and then be evaluated on an unlabeled MNIST-M (see Fig 2).

DANN Model For this session, we will reproduce the model described by Ganin and Lemptisky in their paper "*Unsupervised Domain Adaptation by Backpropagation*", that you can (and are encouraged to) read here.

The model is displayed in Fig 3. The layers in green are the convolutional neural network. Its goal is to learn a good representation for the classifier, whose layers are in blue. However, we want the representation to be domain-agnostic, meaning that in our case, a digit '7' in white, or a digit '7' in purple, have the same representation. To do so, another output is added to the network, represented by the layers in pink. This is the domain classifier branch. Its goal is to tell from the representation produced by the ConvNet if the image comes from the source or target domain.

The important part, is that between the ConvNet in green, and the domain branch in pink, there is a Gradient Reversal Layer (GRL). This layer reverses (i.e. multiplies by a negative number) the gradient. This means that while the domain branch will try to get good at classifying domain, the ConvNet will try to make it impossible to discriminate both domains, and thus will become domain-agnostic.

Partie 2 – Practice

Load the notebook and follow the instructions written in it.

Questions

1. If you keep the network with the three parts (green, blue, pink) but didn't use the GRL, what would happen ?
2. Why does the performance on the source dataset may degrade a bit ?
3. Discuss the influence of the value of the negative number used to reverse the gradient in the GRL.
4. Another common method in domain adaptation is pseudo-labeling. Investigate what it is and describe it in your own words.