Paolo Alberto
Student ID: s278098

Machine Learning and Deep Learning
Politecnico di Torino

Homework 3 report

The purpose of homework 3 is to perform a Domain Adaptation task, implementing the DANN algorithm starting from the pre trained Alexnet model. The idea behind this network is to classify images with a different domain than the one used for training.
The model was trained on a Nvidia GTX 1050, allowing to perform a grid search task that would have been otherwise impractical using Colab.

1 The dataset

In this assignment we will work with the PACS dataset: a collection of nearly 10000 images belonging to 7 different classes and 4 different domains: photos, art paintings, cartoons and sketches. In this assignment we will focus on learning the features of the photo domain, and testing the model on the paintings.

Figure 1: **P**hotos, **A**rt paintings, **C**artoons and **S**ketches dataset

2 Implementing the DANN model

Starting from the original AlexNet model, the new G_d branch is added in the init function of the model, by creating a copy of AlexNet fully connected layers, characterised by the same weights of the original pre-trained network. With these changes in place, the network has now 2 different branches capable of performing classification tasks. The two terminal layers of these branches are then changed by setting the number of output neurons to 7 and 2 respectively.

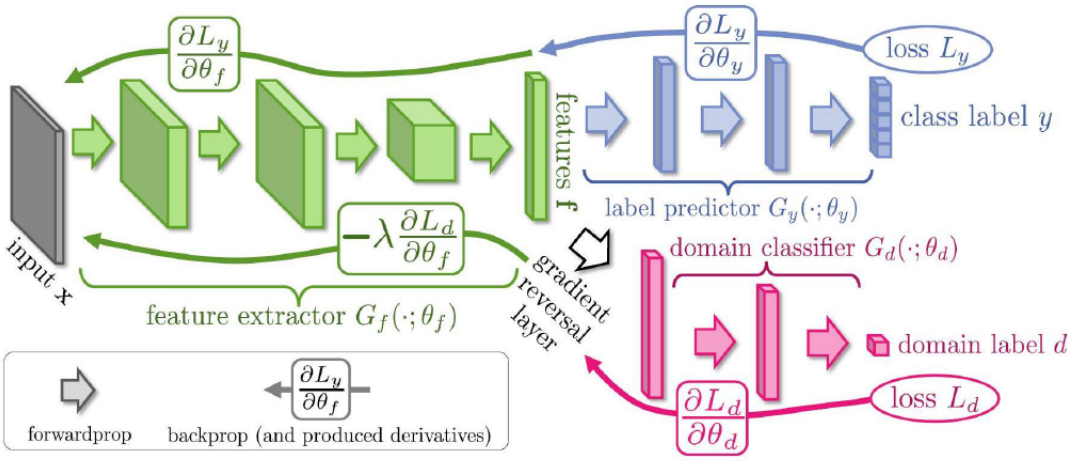

Figure 2: DANN schema

For the first part of the assignment, we will not use the domain classifier G_d yet. Instead, we will implement the classical AlexNet model, and observe the results of the cross-domain classification. After this first implementation, we will use the complete DANN model, observing the changes in accuracy and possible improvements.

3 Training without domain adaptation

The first step is to train the label predictor branch G_y on the photos, and testing it on the paintings, without the help of domain adaptation.
For the first run, the model is trained on 20 epochs, with an initial learning rate of 10^{-3} and a step down policy implemented after the first 10 epochs with gamma $= 0.1$. The accuracy obtained is 47%.

After this first run, some hyperparameter optimization is performed, by tweaking some of the most important parameters (learning rate, epochs, gamma and batch size). Some combinations of these proved to be slightly better, increasing the accuracy by a couple of percentage points, but never going over 50%.
While performing this kind of optimization process, it is important to keep in mind that, when

testing on the Art Paintings, we are essentially cheating, since we are using the same dataset both for the adaptation and the testing phase.

Some of the best combinations of hyperparameters found during this this phase can be found in the table below.

Hyperparameters					Accuracy
LR	Epochs	Step size	Gamma	Batch size	
1e−3	20	10	0.1	128	47%
5e−4	20	10	0.1	64	47.5%
1e−2	20	10	0.1	64	48.2%
5e−3	20	10	0.1	32	**48.5%**

Table 1: Hyperparameter optimization without domain adaptation task

After this optimization process, 48.5% is the highest accuracy obtained on the test set. Keeping in mind these values, we can use now use the best performing hyperparameters on the DANN model that we built in order to perform the actual domain adaptation task.

4 Domain adaptation task

The next step is the implementation of the actual DANN training process. This is done by training the domain classifier branch G_d to learn features characterizing photos and art paintings, and taking advantage of the back propagation of this branch in order to generate some new domain invariant features, that can be used to train on the first domain and test on the second one.
When optimizing the hyperparameters of the DANN model, a new parameter *alpha* needs to be be taken into account. Alpha is a multiplication factor that is applied to the reversed gradients, and represents the weight of the back propagation.

At first, alpha is set to 0.1, and the DANN model is applied using the best performing hyperparameters found before. The resulting accuracy is 51.61%, with respect to the previous value of 48.5% obtained without domain adaptation. In this setting, even a slight improvement like the one observed in this case can be considered as a significative improvement.

A grid search is then performed on α and LR, in order to find the best performing pair of hyperparameters.

LR \ α	0.05	0.1	0.25	0.5
0.0005	48.73%	46.38%	50.48%	47.02%
0.001	49.61%	49.60%	51.26%	18.5%
0.0025	52%	**52.73%**	51.46%	18.5%
0.005	49.85%	51.61%	18.5%	18.5%
0.01	48.38%	18.5%	18.5%	18.5%

Table 2: Grid search results with AlexNet implementation of DANN

Comparing these results with the ones obtained without DANN, we can conclude that a slight increase in accuracy can be obtained by using domain adaptation techniques, if we select the right combination of hyperparameters. As we can see from the grid search table, an high value of α must be combined with a lower learning rate, or else the model will diverge, and the classificator will be useless. We stopped here with the hyperparameter optimization. Some other parameters that could have been tried with the grid search are batch size, epochs and step size

Plotting the confusion matrix for the best performing classifier gives us some interesting insights on the workings of the model.

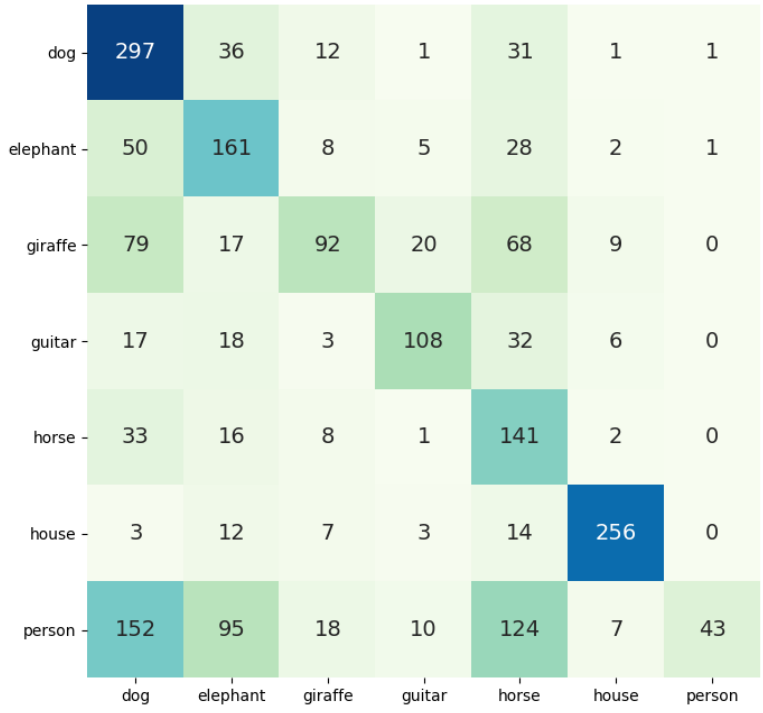

Figure 3: Confusion matrix for the predictions on the art paintings domain

We can see that, when the ground truth is characterized by an image representing a person, the network tends to make more mistakes than usual, often classifying the given input as a dog or a horse. An explanation for this phenomenon could be found in the fact that in the provided images of horses and dogs there are sometimes humans involved in the scene.

Another more plausible explanation would be that the images belonging to the "person" class in the photo domain are exclusively composed of face close-ups, with a lot of similar selfies, often belonging to the same people. The art painting counterparts however, range over a larger spectrum of subjects and actions, making these challenging

By filtering out the "person" class from the accuracy calculations, just to have a feeling of what the results would be in that case, we obtain an accuracy of about 66%.