Homework #3: Domain Adaptation

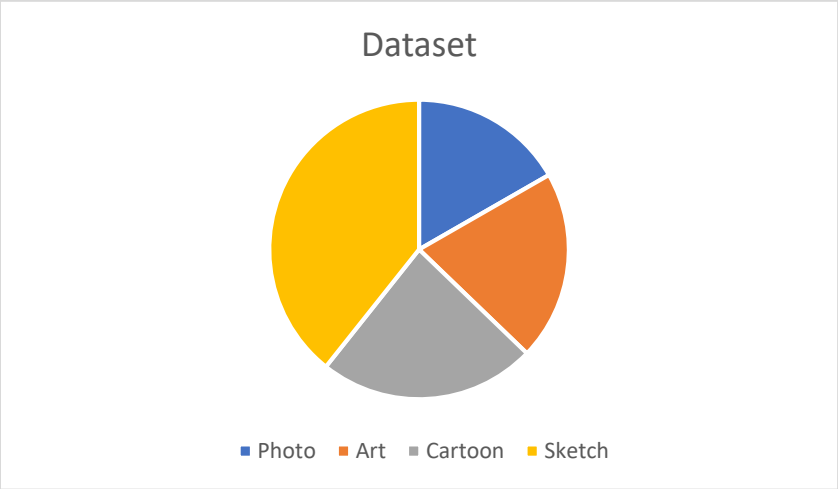
Politecnico di Torino, Simone Dutto

Description of the assignment:

The task is to implement DANN, a Domain Adaptation Neural Network, to work on the PACS dataset using AlexNet as a baseline.
The final result has to be a model which, trained on Photos domain, is able to correctly classify the same classes on Art domain.

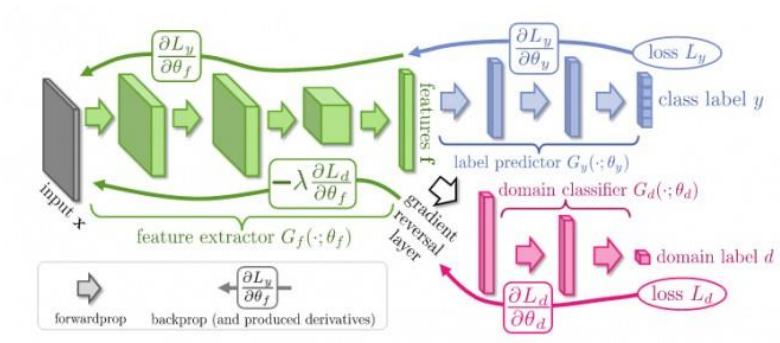
Dataset:

The PACS dataset is divided in 4 main domain classes: Art Painting, Cartoon, Photo, Sketch; and 7 sub-classes for each domain.
The whole dataset has 9991 entries, with these proportions:



Architecture:

The DANN is a rendition of a typical Convolutional Neural Network with an additional branch to push domain invariant features extraction.



The green branch is responsible of feature extraction, the blue branch is used to classify the classes' entries and the pink branch is a binary domain classifier (source/target).

The backpropagation is a little bit different from a typical CNN architecture. In fact, from the pink branch the gradient is reversed, because the features extractor is updated in order to trick the domain classifier and maximize therefore the loss of the domain classificator.

Training:

The train dataset is divided in 2 parts: source (domain label for training) and target (domain label for testing).
The training phase is subdivided in 3 part: first is trained the class label classifier, then the domain classifier both with an entry from source and target dataset. After each step the loss is backpropagated. Besides other hyperparameters this time is crucial to finetune ALPHA, which is the multiplication factor of the reverse gradient.

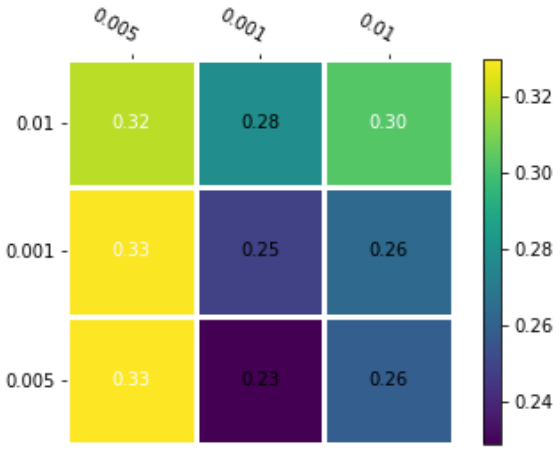
Validation:

Validation is an issue in Domain Adaptation, because tuning the model and seeing the performance on test set is “cheating”, so usually it is used the Cross-Domain Validation.
The validation is done testing the model performances in other domains, the best hyperparameters are then chosen and the model is tested on the target domain.
In my case the validation set is Cartoon and Sketch domains, the accuracy is the average between the 2 accuracies.

To evaluate better the performance, I tested accuracy on Photo -> Art without Domain Adaptation and different Learning Rates.

The accuracy on validation set is up to 30.4% and on test set is 46.4%. Because there are 7 classes the worst model (random output) has 14.3% accuracy, so my model is not good but still able to generalize a bit even without Domain Adaptation.

Then I activate the part relatively to Domain Adaptation and I have applied the same validation technique.



The best hyperparameters are alpha=0.01 and LR=0.01, with an accuracy of 33.0% on validation set.
On test set the best model performs better than the best model without Domain Adaptation, with 52.2% (5.8% plus model-noDA).
We can notice that in general the additional branch tends to higher the accuracy on the validation set, which implies that the model is working well pushing domain invariant features extraction.