DLCV3

Hamann, Clarissa

December 2019

1 Problem 1: GAN

1.1 Architecture and Implementation

Due to time issues I could not start raining my model. The architecture is inspired from [3].

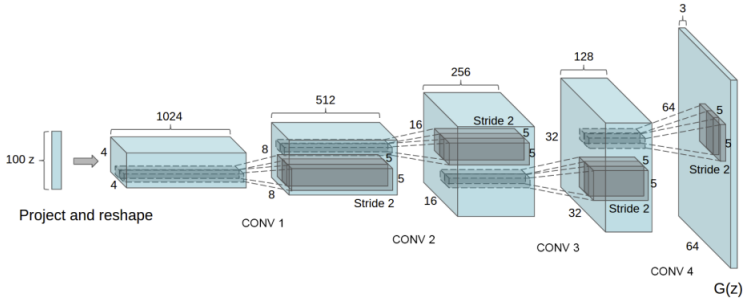

Figure 1: DCGAN

The model has an Generator and a Dicriminator.

1.2 32 Images

1.3 Learnings

The model has an Generator and a Dicriminator. The Generator tries ti generate fake images and the generator has to detect if the image is a fake one or real one. The output is percentage.

2 Problem 2: ACGAN

3 Problem 3: DANN

Sources for code: [1] [4] Source: [2]

3.1 1-3

	SVHN–>MNISTM	MNISTM–>SVHN
trained on source	45.19	33.48
adaption	43.71	44.61
trained on target	94.64	86.87

3.2 4 Visualization

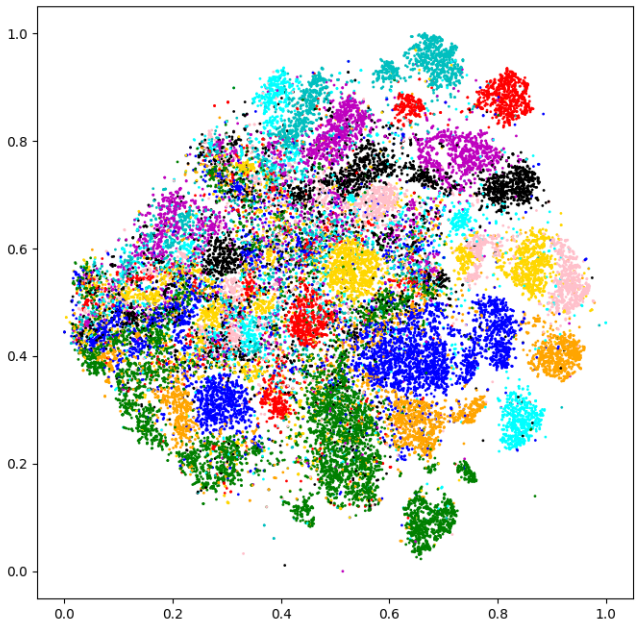

Figure 2: Target: SVHN, classes

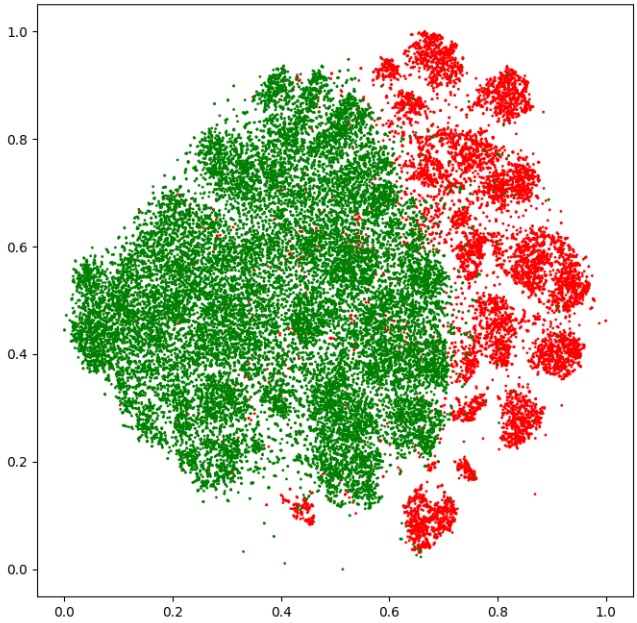

Figure 3: Target: SVHN, classes

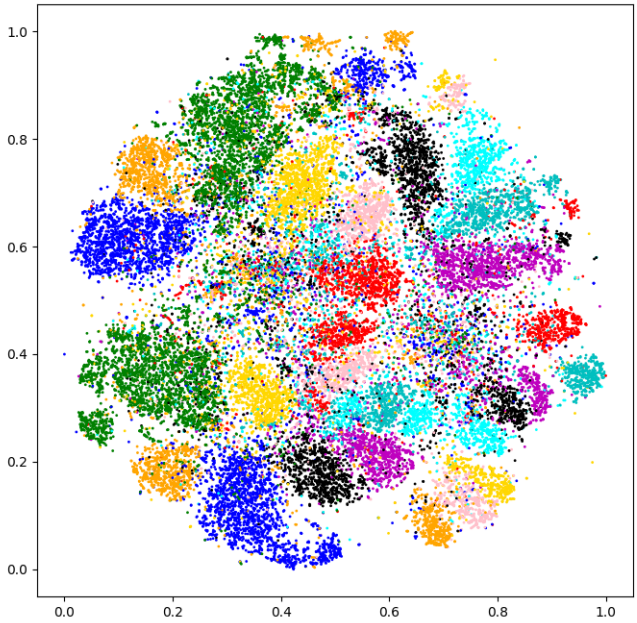

Figure 4: Target: mnistm, classes

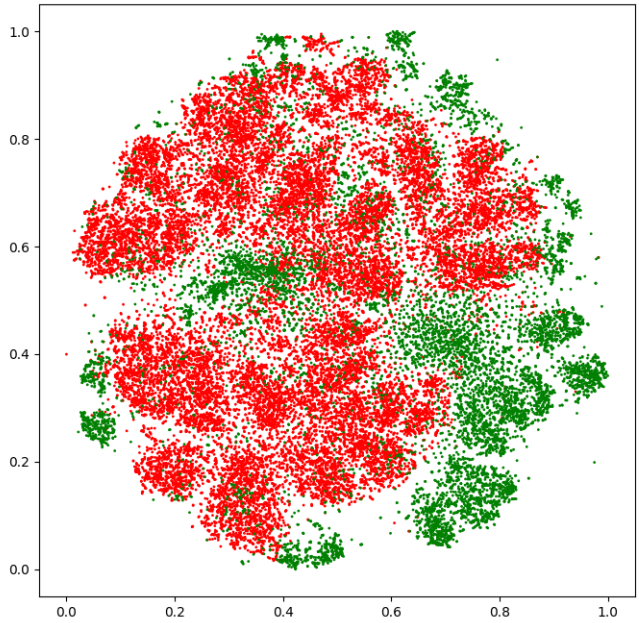

Figure 5: Target: mnistm, classes

3.3 5 Describe the architecture implementation detail of your model

The model has three main parts: the feature extraction, the class classifier and the domain classifier.

Feature extractor:
- 9 layers
Conv2d
Batch Normalization
Max Pooling
ReLU
Conv2d
Batch Normalization
Dropout 2d
Max pooling 2d
ReLU

class classifier:
- 9 layers
Linear
Batch Normalization
ReLU
Dropout 2d
Linear
Batch Normalization
ReLU
Linear
LogSoftMax

domain classifier:
- 5 layers
Linear
Batch Normalization
ReLU
Linear
LogSoftMax

In the forward function at first the feature extractor get trained.The feature get passed to the class classifier. before the get passed to the domain classifier a reverse Layer is added. The forward function has two outputs- the output of the class classifier and the output of the domain classifier. [1]

3.3.1 6

GAN has the purpose to handle a new changed domain of the input data while the task domain (the labels) remained the same. The model get trained with a labeled source domain and a target domain. There are several techniques, here the adversarial-based Domain Adaption is used (DANN). We are generating synthetic data. This synthetic data should be somehow related to the source domain with labels. As you can see in table of part 3.1 if the network get trained with the source domain and get tested with the source domain testset, the results are good. Of course if you test it with the different target domain testset, the result is poor, because the model can not detect similarities in the future. The same happens with a model trained on target data. Tested with the testset of the same domain it works quite well and with the other one (source) not. [2]

4 Problem 4

References

[1] fungtion. GitCode - DANN. https://github.com/fungtion/DANN/blob/master/models/model.py, 2008. [Online; accessed 02-Dec-2019].

[2] Branislav Holländer. Deep Domain Adaptation In Computer Vision. https://towardsdatascience.com/deep-domain-adaptation-in-computer-vision-8da398d3167f, 2008. [Online; accessed 02-Dec-2019].

[3] Nathan Inkawhich.

[4] tengerye. GitCode - pytorch$_DANN, howpublished =$ "$https://github.com/CuthbertCai/pytorch_DANN/blob/master/main.py$", $note =$ "$[online; accessed02 - dec - 2019]$".