Generative Adversarial Networks
Assignment 3
Deep Learning in Computer Vision

June 2019

The main goal of this exercise is to implement a CycleGAN that can convert images of horses to zebras, and vice versa.

MNIST

You have also been working on creating networks that can generate digits from the MNIST dataset.
Feel free to use 1-2 pages of your assignment to: show results from these networks, describe their architectures, the training process, and describe your process of making these choices.

Data

We have supplied you with a dataset for this task. You can find the images on the servers under `/scratch/horse2zebra/`.
The dataset contains two folders: train and test. Inside each of these folders there are two folders: A and B, where A contains pictures of horses, while B contains pictures of zebras. If you wish to have the data on your own machine, we have also uploaded it to the course group on DTU Inside.

CycleGAN

Implement and train a CycleGAN to convert horses to zebras and vice versa, evaluate its performance, and document the process. For details of CycleGAN and a suggested network architecture, see `Lecture_3_2.pdf`, and the CycleGAN paper.

Please **include your code** for the *generators*, *discriminators* and *training* as an appendix, in a smaller, but still legible font.

Things to include in your report could be:

- How is your model performing? Is it able to carry out the desired task? Only on some images? Do you notice common things in the images it does good/bad?

- How is the performance of the discriminator networks compared to the generator? How large are the contributions from each loss term? Does this change over the epochs?

- Try training your discriminator with pictures generated by previous versions of the generator, in addition to the current ones.

    Keep track of the last 50 generated images, and randomly select one of these to also compare against when updating the discriminator. Similar to section 2.3 in this paper

- Training a CycleGAN on different domain translation problem?

- Try using a different architecture for the generator. A U-net could be an idea. How does this affect the results?

- How does a different GAN loss affect the training?