High Performance Machine Learning
Lab 1

Paweł Rościszewski
Department of Computer Architecture
Faculty of Electronics, Telecommunications and Informatics
Gdańsk University of Technology

1 Introduction

This year, in the HPML labs we will investigate and modify a TensorFlow training implementation of StarGAN [1], a generative adversarial approach to image-to-image translation. Since it is a GAN, the training deals with two networks at the same time: the generator and the discriminator. StarGAN introduces a specific methodology for multi-domain robustness, that can be briefly explained by the figure from the original paper [1]:

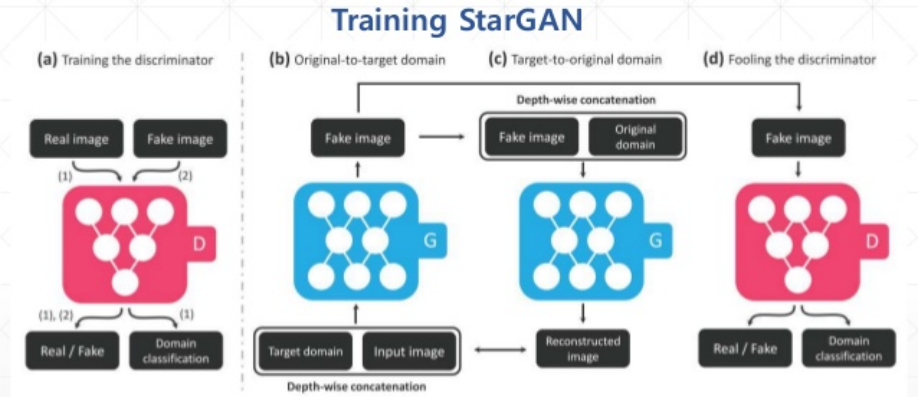

Figure 3. Overview of StarGAN, consisting of two modules, a discriminator D and a generator G. **(a)** D learns to distinguish between real and fake images and classify the real images to its corresponding domain. **(b)** G takes in as input both the image and target domain label and generates an fake image. The target domain label is spatially replicated and concatenated with the input image. **(c)** G tries to reconstruct the original image from the fake image given the original domain label. **(d)** G tries to generate images indistinguishable from real images and classifiable as target domain by D.

The tasks to accomplish within the HPML lab will focus on implementation and performance issues, and not on the quality of the trained generative models. However, the students are encouraged to modify the training parameters and architecture to learn also about the details of GAN approaches.

2 Task 1: Run the example (6 points)

In the laboratory resources, the students will find a StarGAN training program prepared in a form of a Jupyter notebook. To start working with the code, create your own copy of the sample:

```
> cp -r /home/macierz/roy/hpmllab/ hpmllab
> cd hpmllab
```

In order to run the code, a Python3 environment will be needed with all required dependencies. As described in the Lab introduction, the students are encouraged to use *virtualenv* and install the dependencies using the prepared requirements file:

```
> ...prepare or copy the virtual environment (see Lab introduction)
> source venv/bin/activate
> pip install -r requirements.txt
```

After starting a Jupyter notebook in the hpmllab directory (see Lab introduction), the HPMLLab.ipynb notebook should be available. The first task in our lab is to find missing parts in the code and fill them in in order to run the code successfully. Image domains for the image-to-image translation problem will have to be chosen (see [2]).

Additionally, test data has to be provided for model evaluation, consisting of three test images of 128x128 dimensions. The images should contain human faces that will be (hopefully) translated to a different domain. It might be fun to use a photo of yourself, a friend, relative or a favorite person.

To pass the task: present preliminary results of a running training application.

3 Task 2: Check memory usage (4 points)

Machine learning engineers should be aware of the hardware resources used by their programs. One of the crucial resources in High Performance Computing is memory usage. In this task, let's examine what is the usage of GPU memory of our program.

By default, the TensorFlow library allocates almost all GPU memory, which makes it hard to observe how much memory is actually needed. However, it is possible to configure the sessions, so that they allocate memory dynamically when it is needed. It is done by setting the allow_growth flag in the ConfigProto object.

```
config.gpu_options.allow_growth = True
```

The task is to research online how this parameter should be set in the Estimator API that is used in the sample code and test it in practice.

*N*ote that in order to make the configuration changes happen, we will need to free the memory currently allocated by TensorFlow. One way to do this is restarting the computational kernel in the Jupyter notebook.

To pass the task: show actual GPU memory usage in the nvidia-smi output during the training.

References

[1] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation," *arXiv:1711.09020 [cs]*, Sept. 2018. arXiv: 1711.09020.

[2] L. Celona, S. Bianco, and R. Schettini, "Fine-Grained Face Annotation Using Deep Multi-Task CNN," *Sensors*, vol. 18, p. 2666, Aug. 2018. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.