High Performance Machine Learning
Lab introduction

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

1 Technicalities

In this edition, the HPML laboratory will be organized as a remote class. Each student should be assigned a number of desXX machine (from 01 to 18). The laboratory hosts are available via SSH using the corresponding IP address: 172.20.83.2XX. These IP addresses are reachable only through the Faculty network, so one of the two following access methods is suggested:

- use the firewall kask.eti.pg.gda.pl as a proxy; use the KASK account for login; to access the Jupyter notebook via HTTP, use SOCKS proxy (ssh -D)[1]
- use the Faulty OpenVPN virtual private network[2]

2 Introduction

The goal of these laboratory classes is to increase practical skills of the students by engaging them in programming tasks for machine learning, related to the performance-oriented concepts discussed during the lectures. The TensorFlow framework has been chosen as a base for implementing these tasks. In this lab we will focus on a MSG-GAN training program.

In the GAN approach, actually two separate deep neural networks (DNNs) are trained together, competing each other: the generator and the discriminator. The generator tries to generate fake images in such a way, that the discriminator is unable to distinguish if they are real. The discriminator tries to distinguish this, but shares training information with the generator, so both of them can gradually improve. More information about the GAN approach can be found in the original paper[3] by Ian Goodfellow et al. and in many tutorials online.

[1] https://linuxize.com/post/how-to-setup-ssh-socks-tunnel-for-private-browsing/
[2] http://starter.eti.pg.gda.pl/openvpn/
[3] https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

In this lab we will focus on the performance-oriented aspects of the training program. Computational performance is crucial here, since there are two deep models trained together and usually multiple training steps are needed to achieve a decent generator.

During the laboratory classes, we will communicate using MSTeams within the dedicated HPML team (link shared via eNauczanie forum). A separate channel is created for each laboratory group, where specific lab schedule can be found. At the end of the semester, we will have two additional terms that can be shared by all the groups to give the possibility catch up if somebody missed a class or struggled with a task.

There will be six basic laboratory terms. Before each of them, a specific lab instruction will be published on eNauczanie. Each of the six laboratory tasks will be worth 10 points, giving the total number of points equal to 60. This means that 30 points will be required to pass the task. Each task should be solved during the appropriate term, while there will be a possibility to get half of the points remaining in the task in the following lab term.

Reporting solutions will be done by sharing a link and access key to a Jupyter notebook and discussing the solutions in MSTeams. In order to run the Jupyter notebook, we need to set up a Python3 environment with all required dependencies. For this, we will use *virtualenv*:

```
> virtualenv -p python3 venv
> source venv/bin/activate
> pip install jupyter
```

Next, let's start the Jupyter notebook:

```
> jupyter notebook --ip 172.20.83.2XX
```

In the notebook output we should see lines similar to the following:

```
[I 10:45:22.197 NotebookApp] Serving notebooks from local directory: ...
[I 10:45:22.197 NotebookApp] The Jupyter Notebook is running at:
[I 10:45:22.197 NotebookApp] http://172.20.83.2XX....
[I 10:45:22.197 NotebookApp]  or http://127.0.0.1:...
[I 10:45:22.197 NotebookApp] Use Control-C to stop this server \
and shut down all kernels (twice to skip confirmation).
```

Let's use a browser to access the notebook using the URL given in the Jupyter notebook output (the one that uses the appropriate IP).