Introduction to Deep Learning (IN2346)

Technical University Munich - WS 2019/20

1. Python Setup

Prerequisites: - Unix system (Linux or MacOS) - Python version 3.7.x - Terminal (e.g. iTerm2 for MacOS) - Integrated development environment (IDE) (e.g. PyCharm or Sublime Text)

For the following description, we assume that you are using Linux or MacOS and that you are familiar with working from a terminal. The exercises are implemented in Python 3.7. Note that you might be unable to install some libraries required for the assignments if your python version $<$ 3.7. So please make sure that you install python 3.7 before proceeding.

If you are using Windows, the procedure might slightly vary and you will have to Google for the details. We'll mention some of them in this document.

To avoid issues with different versions of Python and Python packages we recommend to always set up a project specific virtual environment. The most common tools for a clean management of Python environments are *pyenv*, *virtualenv* and *Anaconda*.

In this README we provide you with a short tutorial on how to use and setup a *virtuelenv* environment. To this end, install or upgrade *virtualenv*. There are several ways depending on your OS. At the end of the day, we want

```
which virtualenv
```

to point to the installed location.

On Ubuntu, you can use:

```
apt-get install python-virtualenv
```

Also, installing with pip should work (the *virtualenv* executable should be added to your search path automatically):

```
pip3 install virtualenv
```

Once *virtualenv* is successfully installed, go to the root directory of the i2dl repository (where this README.md is located) and execute:

```
virtualenv -p python3 --no-site-packages .venv
```

Basically, this installs a sandboxed Python in the directory `.venv`. The additional argument ensures that sandboxed packages are used even if they had already been installed globally before.

Whenever you want to use this *virtualenv* in a shell you have to first activate it by calling:

```
source .venv/bin/activate
```

To test whether your *virtualenv* activation has worked, call:

```
which python
```

This should now point to `.venv/bin/python`.

From now on we assume that that you have activated your virtual environment.

Installing required packages: We have made it easy for you to get started, just call from the i2dl root directory:

```
pip3 install -r requirements.txt
```

The exercises are guided via Jupyter Notebooks (files ending with `*.ipynb`). In order to open a notebook dedicate a separate shell to run a Jupyter Notebook server in the i2dl root directory by executing:

```
jupyter notebook
```

A browser window which depicts the file structure of directory should open (we tested this with Chrome). From here you can select an exercise directory and one of its exercise notebooks!

Note:For windows, use miniconda or conda. Create an environment using the command:

```
conda create --name i2dl python=3.7
```

Next activate the environment using the command:

```
conda activate i2dl
```

Continue with installation of requirements and starting jupyter notebook as mentioned above, i.e.

```
pip install -r requirements.txt jupyter notebook
```

2. PyTorch Installation

In exercise 3 we will introduce the *PyTorch* deep learning framework which provides a research oriented interface with a dynamic computation graph and many predefined, learning-specific helper functions.

Unfortunately, the installation depends on the individual system configuration (OS, Python version and CUDA version) and therefore is not possible with the usual `requirements.txt` file.

Use this wheel inside your virtualenv to install *PyTorch*:

OS X

```
pip install torch==1.3.0 torchvision==0.4.1
```

Linux and Windows

“‘

CUDA 10.0

```
pip install torch==1.3.0 torchvision==0.4.1 -f https://download.pytorch.org/whl/torch_stable.html
```

CUDA 9.2

```
pip install torch==1.3.0+cu92 torchvision==0.4.1+cu92 -f https://download.pytorch.org/whl/torch_stable.h
```

CPU only

pip install torch==1.3.0+cpu torchvision==0.4.1+cpu -f https://download.pytorch.org/whl/torch_stable.htm
“‘

3. Exercise Download

The exercises would be uploaded to Moodle. You need to login with your moodle account and download the exercises from there. At each time we start with a new exercise you have to populate the respective exercise directory.

The directory layout for the exercises

```
i2dl_exercises
   datasets                # The datasets required for all exercises will be downloaded 
   exercise_0
   exercise_1
   exercise_2
   exercise_3
   LICENSE
   README.md
```

4. Dataset Download

To download the datasets required for the exercise, navigate to the exercise directory {0,1,2,3} and run the corresponding `download_dataset.sh` script by executing `./download_dataset.sh` depending on your OS. All the datasets will be downloaded in the `datasets` folder located in the `i2dl_exercises` directory.

If you are unable to download the dataset from the scipts provided, then you can manually download it from this link. For windows use this link and place it in the appropriate directory structure as shown below.

You will need ~400MB of disk space. The directory structure for datasets downloaded is like:

```
i2dl_exercises
       datasets                   # The datasets required for all exercises will be downloa
```

```
        cifar10                # Dataset directory
            cifar10.p          # dataset files
```

5. Exercise Submission

Your trained models will be automatically evaluated on a test set on our server. To this end, login or register for an account at:

https://dvl.in.tum.de/teaching/submission/

Note that only students, who have registered for this class in TUM Online can register for an account. This account provides you with temporary credentials to login onto the machines at our chair.

After you have worked through an exercise, you need to compress the exercise directory with `zip` extension(not `rar`) and submit it to the submission server(should not include any datasets). All your trained models should be inside `models` directory in the exercise folder. You can use `create_submission.sh` to create your zip file, it will create a zip of only required files and folders. In order to create the zip, just execute:

```
./create_submission.sh
```

For windows, check the files the shell script zips (models directory, exercise code directory and all ipynb files, while keeping the structure same) and create that zip manually.

You can login to the above website and upload your zip submission server. Once uploaded you can see the all the models you submitted (if there's a blank page for a long while, refresh). Also, you can select the models for evaluation.

You will receive an email notification with the results upon completion of the evaluation. To make it more fun, you will be able to see a leaderboard of everyone's (anonymous) scores on the login part of the submission website.

Note that you can re-evaluate your models until the deadline of the current exercise. Whereas the email contains the result of the current evaluation, the entry in the leader board always represents the best score for the respective exercise.

6. Acknowledgments

We want to thank the **Stanford Vision Lab** and **PyTorch** for allowing us to build these exercises on material they had previously developed.