Installing TensorFlow on Windows

Homework assignments are mandatory. In order to be granted with 8 ECTS and your grade, you must pass 9 out of 10 homeworks and correct the homework of another group every week. Upload your solution as iPython notebook (*.ipynb*) and as HTML export until the following Saturday 23:59 into the public *Homework Submissions* folder on studip. Homework assignments start next week. This week there is nothing you have to upload.

If you encounter problems, please do not hesitate to send your question or concern to lbraun@uos.de. Please do not forget to include [TF] in the subject line.

1 Sign up to a group on studip

Please visit the *Teilnehmende* section and click on *Gruppen* in the sub menu on the left side of the course page on studip and sign up in one of the groups **until Monday October 30th. 23:59**. Each group must consist of three members.

This is the group your are going to hand in your weekly homework assignments and *Final Task* with.

2 Bias and algorithms

In this week's lecture we talked about how we can use an artificial model of neurons to approximate a function, which maps some input to some output. Watch Cathy O'Neil's Ted talk The era of blind faith in big data must end, where she points out pitfalls and the inherent biases, that you easily overlook when working with machine learning algorithms.

3 Installing TensorFlow

The following instructions are an extended version of the official installation guidelines. There are two different versions of TensorFlow, a GPU and a CPU version.

In order to install TensorFlow, you need to have a 64-bit operating system.

A single backslash *continues a command on the next line* in Bash. You can drop them, when you type the commands into your terminal. I just used them because some of the commands do not fit in a single row in this file.

For example

```
$ echo \
123
```

is the same as

```
$ echo 123
```

3.1 Installing Anaconda

We are going to use *Anaconda* in this course. *Anaconda* is a program to create and manage virtual environments on your computer, which contain their own python instance and python libraries, separated from the one already installed on your computer. This has the advantage, that there will be no conflicts with already existing, maybe outdated versions of python and python libraries on your computer. Further, by using *Anaconda*, all members of the course are going to use the same version of python and TensorFlow.

Please download the **python 3.6 version** of Anaconda from the Anaconda homepage and follow the installation instructions provided on the same page.

There is no need to install further packages like *numpy* or *matplotlib*, since *Anaconda* contains current versions of them already.

3.2 Create a new conda environment

Next, we are going to create a new *conda environment* called *tensorflow*. Open a terminal and type the following command:

```
$ conda create -n tensorflow python=3.5
```

3.3 Activating and deactivating the conda environment

In order to enter the newly created virtual environment, you can use the following Bash command:

```
$ source activate tensorflow
```

In order to leave it again use:

```
$ source deactivate
```

Always make sure to activate the conda environment before you start working on your TensorFlow code.

3.4 GPU Support

Before we can start to download and install the TensorFlow package, we have to find out if your GPU supports CUDA. Executing TensorFlow code on your GPU rather than on your CPU can speed up the training of your network drastically.

Open the *Windows Display Settings* to find out the exact name of your grphic card. If your computer is not equipped with a *NVIDIA GPU*, you can continue with *Intalling the CPU version of TensorFlow* on page 5. Otherwise, visit the list of supported GPUs and search for your GPU. If the *compute ability version* (first column of the table) is smaller than 3, then your GPU is to old and you can continue with *Installing the CPU version of TensorFlow* too. **Only if you have a supported NVIDIA GPU**, follow the next steps to install the *CUDA* toolkit and *cuDNN*.

3.4.1 Installing the CUDA toolkit

Download and install version 8.0 of the toolkit from the CUDA download page.

3.4.2 Installing cuDNN

In order to download cuDNN you have to register and join the *Accelerated Computing Developer Program*. Download and install version 6 of cuDNN from the cuDNN homepage.

3.5 Adjusting the %PATH% variable

As described in the NVIDIA documentation, you need to add both, the relevant Cuda pathnames and the cuDNN installation path to your %PATH% variable.

3.6 Installing TensorFlow with GPU support

Next, activate your *tensorflow* conda environment:

```
$ source activate tensorflow
```

and install TensorFlow by executing the following comand:

```
(tensorflow)$ pip install --ignore-installed \
--upgrade tensorflow-gpu
```

You can now proceed with the instructions to *Test run TensorFlow* on the last page of this document.

4 Installing the CPU version of TensorFlow

Activate your *tensorflow* conda environment:

```
$ source activate tensorflow
```

and install TensorFlow by executing the following comand:

```
(tensorflow)$ pip install --ignore-installed\
--upgrade tensorflow
```

You can now proceed with the instructions to *Test run TensorFlow*.

5 Test run TensorFlow

Make sure to be in the activated *tensorflow* conda environment. Then, start a new python session by executing:

```
(tensorflow)$ python
```

Next, import the TensorFlow library into your python session and execute the following *Hello World* program:

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
Hello, TensorFlow!
```

Congratulations, you successfully installed TensorFlow!