

Homework 1: Setup

Implementing ANNs with TensorFlow

This homework is not mandatory, but meant to be a preparation for the course. It includes setting up your machine for implementing and training artificial neural network and setting up you for understanding the concepts that will be covered in the lecture.

I want to be clear that, although there are no hard requirements for being enrolled in this course, this course presupposes knowledge in coding and math. In this homework I will try to summarize these presuppositions. It is divided in three parts:

1. Setup Work Environment
2. Jupyter Notebooks, Python, NumPy
3. Basics in Linear Algebra and Calculus

1. Setup Work Environment

- a) I recommend using [Google Colab](#). It offers anything you will need for the course. No need to install anything on your computer. All computations are run in the cloud.
Disadvantages: You need a google account. You can't program, if your are offline.
If you want to work in Colab I recommend to visit the first tutorial on 6th November 2019.
- b) You can also install everything on your own machine. Depending your operating system this might be more or less a trouble (Ubuntu usually easy, MacOS often okay as well, Windows can be quite painful).

Because of the different operating systems and different issues that can arise, I will only give a high-level instruction.

1. Install **Anaconda**. It will help you in managing your python packages and installs most important packages all at once. ([Link](#))
2. If you want you can create an **environment** for the course. Environments allow you to install different versions of packages at the same time. ([Link](#))
Note: You might need to reinstall Jupyter Notebook inside your environment in this case.
3. Install **TensorFlow 2.0** with pip. ([Link](#))
Note: In case you have a supported graphics card in your machine you might want to install the GPU version of TF (can make your computations faster) ([link](#)).
For TF 2.0 there is currently no stable version though!

4. Test whether your installation was successful.
 - Open a Jupyter Notebook. (Type `jupyter notebook` in terminal.)
 - Type:
 - `import tensorflow as tf`
 - `print(tf.__version__)` (should print '2.0.0')

2. Jupyter Notebooks, Python, NumPy

I presuppose that you have basic coding skills in Python and you know how to use Jupyter Notebook.

Answer the following questions about working in Jupyter Notebook.

- a) What is a cell? How to code in a cell? How to execute a cell?
- b) How do you add a new cell and remove a cell?
- c) How do you see whether a cell is still running?
- d) How to stop a running cell?/How to interrupt the kernel?
- e) How to re-run all cells from the beginning?
- f) How to change a cell from code to markdown to write in a cell?
- g) Are there keyboard shortcuts for the above mentioned operations? (Might differ between local jupyter notebook and google colab.)

Now that you understand how to work inside a Jupyter notebook you can start coding in python. If you have no experience at all it might help to start with the official tutorial. ([Link](#))
As a guidance: you should be able to understand and use the following concepts:

- a) `if-else-statements`
- b) `for-loops`, `while-loops`
- c) `break`, `continue`
- d) `lists` (+`list-comprehension`)
- e) `functions`
- f) `classes`

Last but not least you will need some basic understanding of NumPy (a library for mathematical computations in python). TensorFlow has a lot in common with it. Again you can have a look at the official tutorial page ([Link](#)).

As a guidance you can try to answer the following questions:

- a) What is a numpy array?
- b) How to create an array out of a list? How to create one filled with 0s?
- c) What does `np.arange()` do?
- d) How to access the shape of an array?
- e) How to reshape an array?
- f) What does it mean to reshape an array with shape (3,2,4) to shape (2,-1)?
- g) How to access an element in an array?
- h) How to access a column/row?

- i) How to access a specific slice of an array? (E.g. second to forth column in the third and fifth row?)
- j) How to perform matrix multiplication?
- k) How to perform the scalar/dot product?

3. Basics in Linear Algebra and Calculus

Finally to understand the theory behind neural networks you will need basic understanding of concepts in linear algebra and calculus. The following is a list of questions and concepts that you should be able to answer. Some questions are more advanced and will be covered throughout the lecture. Those will be marked with a (+).

Calculus

- a) What is the derivative of a function?
- b) What are some simple differentiation rules?
- c) What is the chain rule?
- d) How do you compute the minimum/maximum of a function?
- e) What are global and local minima/maxima?
- f) What is a saddle point?
- g) What is a high-dimensional function?
- h) How to differentiate a high-dimensional function? (+)
- i) What is a partial derivative? (+)
- j) What is the gradient? (+)
- k) What is the jacobian matrix? (+)

Linear Algebra

- a) What is a vector?
- b) What is a matrix?
- c) What is a matrix multiplication? -p
- d) When can two matrices be multiplied?
- e) How are two matrices multiplied?
- f) What is the dot product of two vectors?
- g) Is the dot product a matrix multiplication?
- h) How do you transpose a matrix?