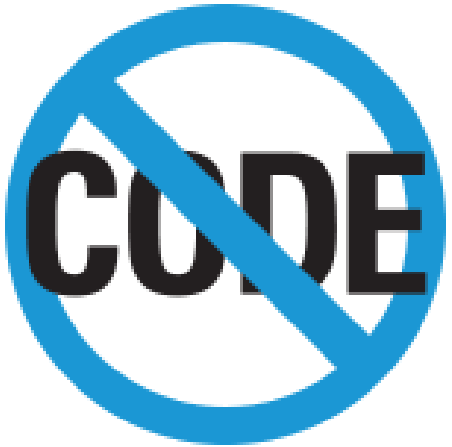
Deep Learning for Computer Vision

Organizational Stuff and Introduction to Exercise 1

Overview

- Why should I do the exercises ?
- Our „tech-stack“
- How to get the exercises ?
- Outlook exercise 1 and upcoming exercise schedule

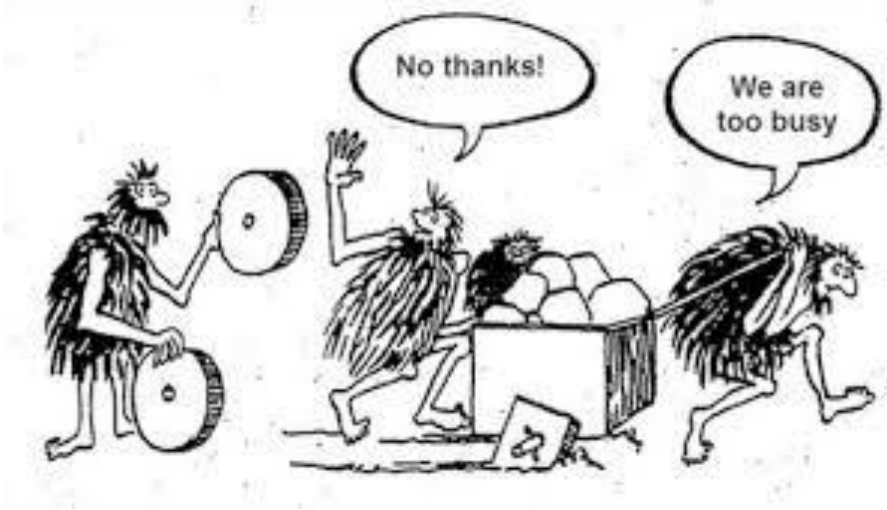


Why should I do the Exercises?

- Bonus points:
  - 3 Exercises, with subparts (3+2+3 = 8 subparts)
  - If 7/8 submissions perform above the threshold
- To get more in-depth knowledge of
  - Models and specific layers
  - PyTorch
- Actual practical experience

Your task for the first two exercises

- Implementation of **classification** pipeline using
  - Traditional machine learning methods
  - Neural Networks
    - Layers
    - Optimizers
    - Etc.
- „Re-invent the wheel“



Later: PyTorch

- Machine Learning library for Python
- GPU support
- Easy, interactive sessions and computation graphs like vanilla python

Submissions - Overview

- S1 Linear classifiers
- S2 Two-Layer NN
- S3 Features

Begin: 14.11.

End: 04.12. 18:00

- S4 Build your own DL library & Classification
- S5 (Regression for House Prices)

Begin: 05.12

End: 18.12. 18:00

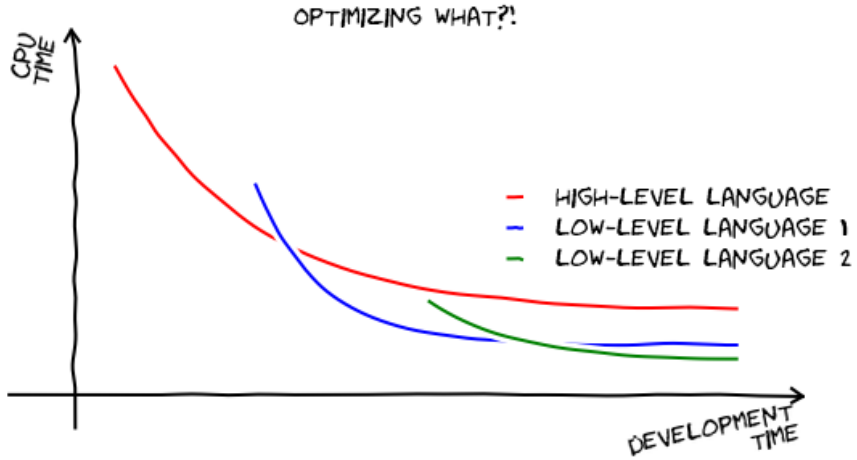
- S6 Pytorch Clasification
- S7 Semantic Segmentation
- S8 Recurrent NN

Begin: 19.12.

End: 22.01. 18:00

Python

- Why python:
  - Very easy to write development code thanks to an intuitive syntax
  - A plethora of inbuilt libraries, esp. for deep learning
- Our stack:
  - Jupyter notebooks
  - Numpy



Jupyter Notebooks

jupyter spectrogram (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Markdown CellToolbar

Simple spectral analysis

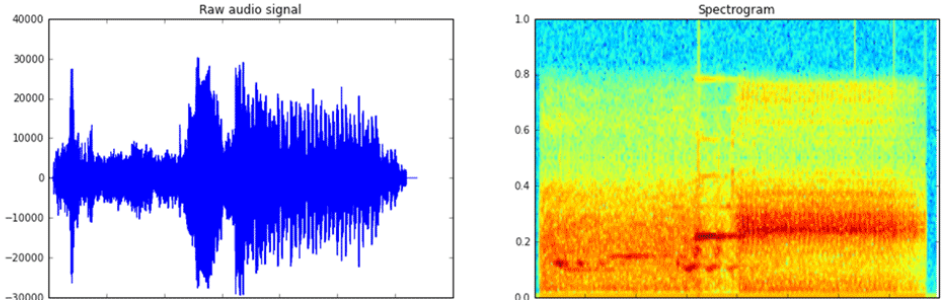
An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n exp^{\frac{-2\pi i}{N} kn} \quad k = 0, \ldots, N-1$$

In [2]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

In [5]:
```
fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```



Numpy

- Support for large, multi-dimensional arrays and matrices
- Powerful operations, e.g., set all non-negative elements of matrix A of shape 1000^2 to zero:

```
for i in range(1000):
    for j in range(1000):
        if A[i][j] < 0:
            A[i][j] = 0
```

Link for everyone:
http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-2-Numpy.ipynb

Numpy

- Support for large, multi-dimensional arrays and matrices
- Powerful operations, e.g., set all non-negative elements of matrix A of shape 1000^2 to zero:

```
A[A<0] = 0
```

Link for everyone:
http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-2-Numpy.ipynb

How to get the exercises

- Main parts:
  - Moodle
    - Distribution of exercise code and datasets
    - Download **exercise_0.zip** to get started
  - Submission webpage: https://dvl.in.tum.de/teaching/submission/
    - Overview of your current submission status
    - Leaderboard

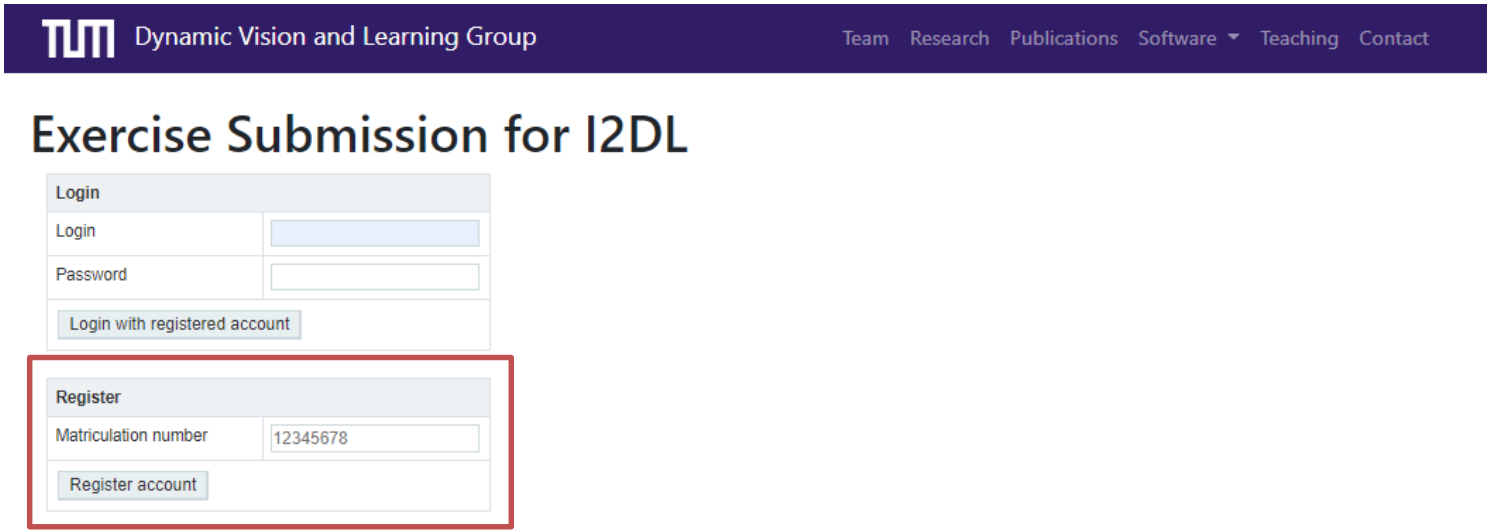
Python Setup

- New users: install python3
  - `README.md`
- „Advanced" users:
  - Virtual environment or anaconda
    - `pip install -r requirements.txt`
  - Regular system python (not recommended)
    - `pip install -r requirements.txt`

New python users: http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-1-Introduction-to-Python-Programming.ipynb

How to submit exercises

- Register at our submission webpage (https://dvl.in.tum.de/teaching/submission/)

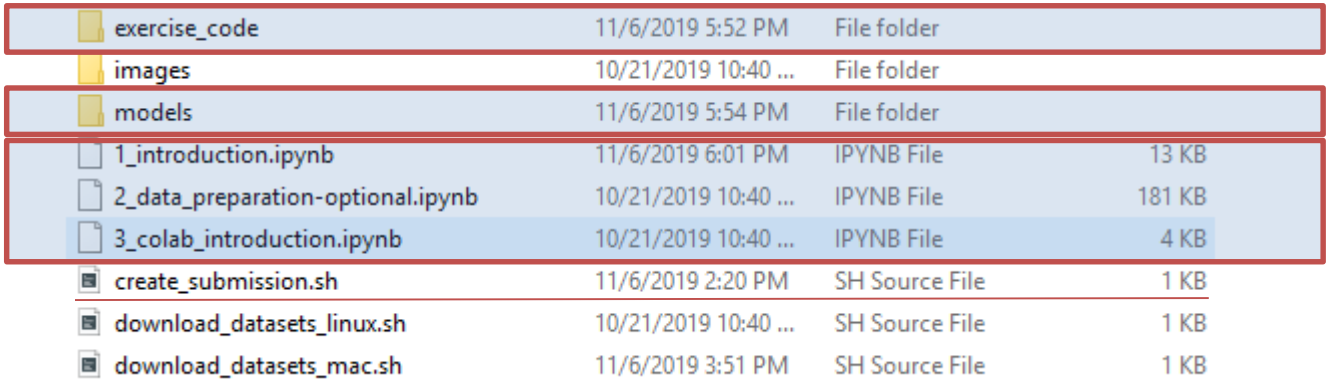


How to submit exercises

- Register at our submission webpage (https://dvl.in.tum.de/teaching/submission/)
  - Sign up with valid matriculation number
  - Get `id` and `password` via mail from tum-online (it will display the email address)

How to submit exercises

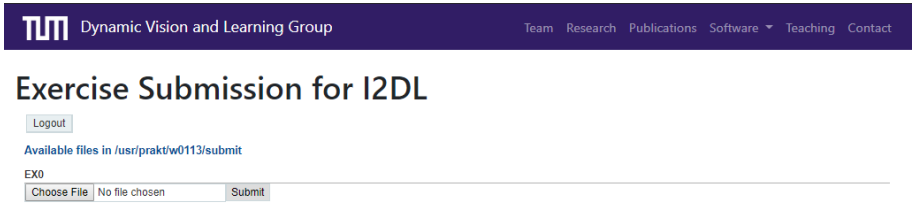
- Submit code + model with
  - Upload code + model for respective exercise by zipping them (no rar, gzip)
  - Note: You will submit your whole code folder as well as your trained models



exercise_0.zip

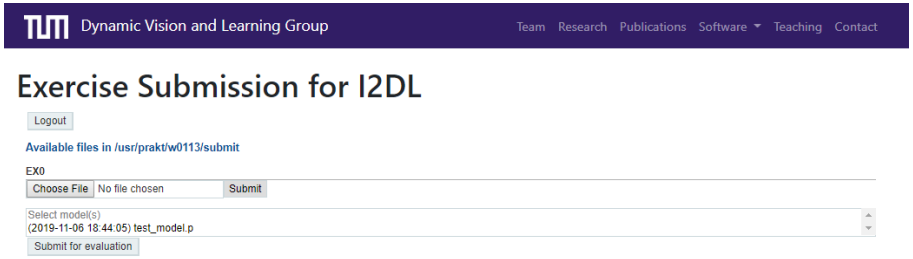
How to submit exercises

- On the submission webpage - upload the zip



How to submit exercises

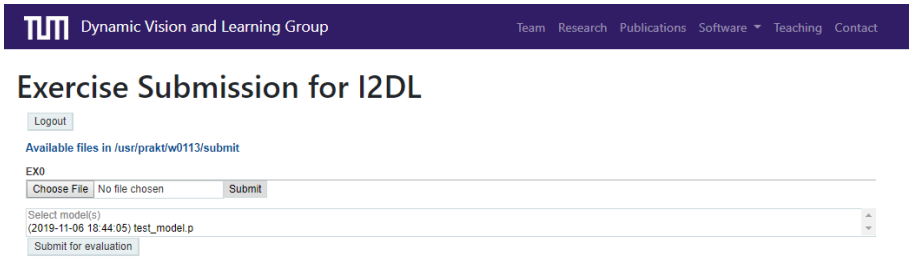
- On the [submission webpage](#) - upload the zip



  - Select the model you want to evaluate

How to submit exercises

- On the submission webpage - upload the zip

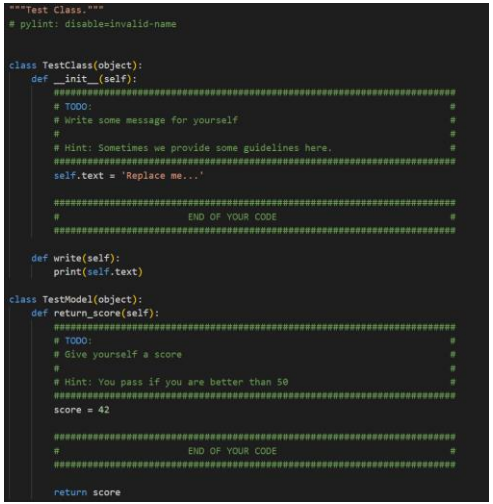


- Wait for email with your score
  - Uses TUM email
  - Refresh webpage until it no longer says „job currently waiting in queue" and check out your score the

Live Demo

Exercises FAQ

- I don't want to code in notebooks. Can I use my favourite IDE?
  - Yes
- Cool, so I can just change the whole code structure?
  - No



Exercises FAQ

- I don't want to code in notebooks. Can I use my favourite IDE?
  - Yes
- Cool, so I can just change the whole code structure?
  - No
  - You can write any helper functions, but keep the skeleton classes intact (i.e., don't rename important functions or variables)
  - You will upload all files and those will be archived on our end

Threshold and Submission FAQ

- How do I know that I passed?
  - Once you submit a score that surpasses the threshold, you will receive an email that contains a message which tells you that you passed this submission
- Help, I got this message a second time!?
  - You will receive this message every time you submit an exercise that exceeds the score

Threshold and Submission FAQ

- I submitted another model which was below the threshold. Do I have to resubmit the old model?
  - No, once one models surpasses the threshold, you are done with this submission (for the bonus)

- How long does it take to do a submission?
  - Depends on the submission, however you will have to train models for all submissions which can take hours to run on your machine, regardless how much coding you have to do

Exercise 0

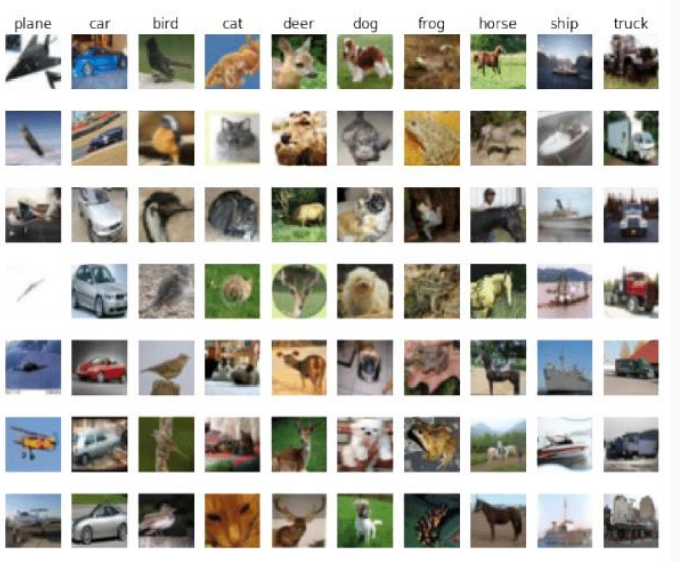
- Dummy exercise – not graded
- Just for getting familiar with the system and for setting up the dev-environment
- Available on moodle

Exercise 1

- Starts in one week (14.11.19)
  - Exercise 1 zip will be uploaded to moodle
  - Move it into your main i2dl folder
  - 3 weeks in total
- Submissions:
  - Softmax classification
  - Two layer neural network classification
  - Feature classification

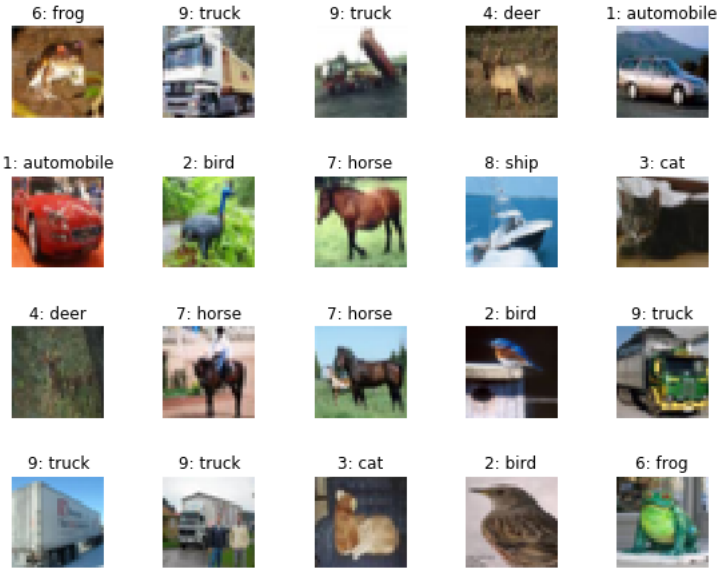
Exercise 1

- Data Pipeline
  - Reading input and visualizing
  - Splitting data in train and test
  - Normalizing



Exercise 1

- Implementation of **classification** of CIFAR 10 images



Submission 1:

Softmax Classifier

Submission 2:

Two Layer Neural Network

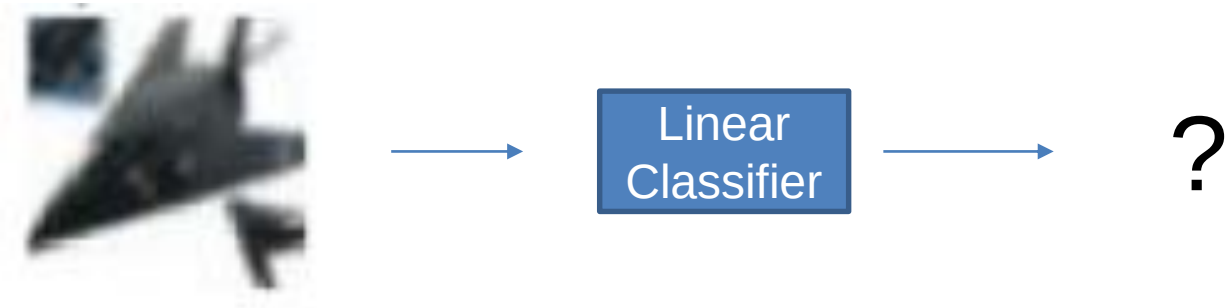
Submission 3:

Feature Neural Network

DEADLINE:
Wed, 04.12.19, 18:00pm

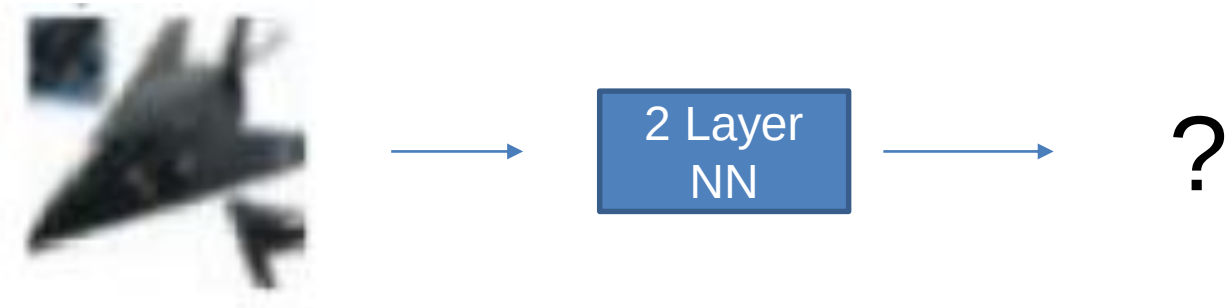
Exercise 1

- Softmax Classifier



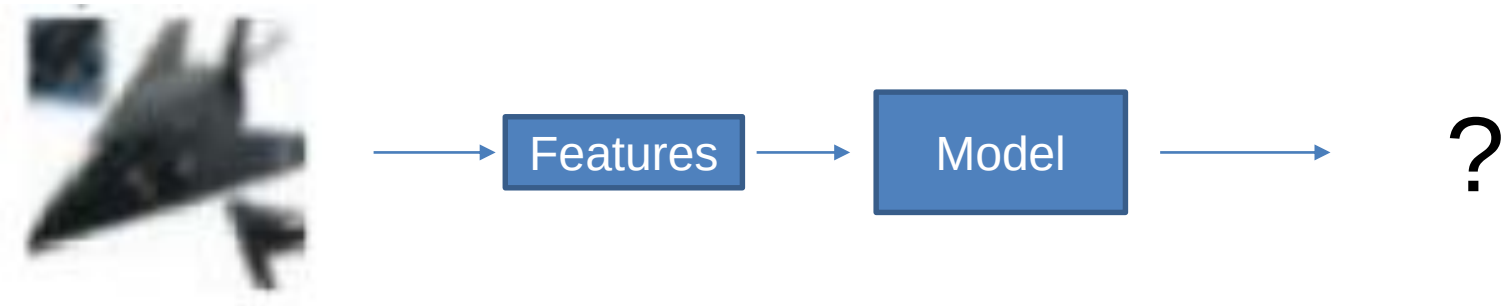
Exercise 1

- Two layer NN



Exercise 1

- Feature Classification



Exercise 1

- Detailed description in notebook

No exercise session the next weeks!
Next one on 5th December
(Exercise 1 will start without session)

Links

- CS231n turials: http://cs231n.github.io
- Scientific Python Introduction/Lectures: https://github.com/jrjohansson/scientific-python-lectures
- Math background: http://parrt.cs.usfca.edu/doc/matrix-calculus/index.html
- I2DL README.md on moodle/the i2dl git

Questions?