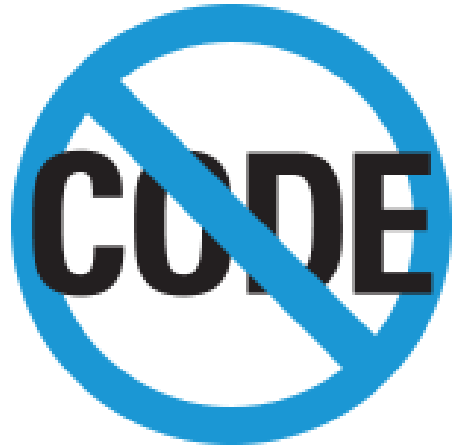


Deep Learning for Computer Vision

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



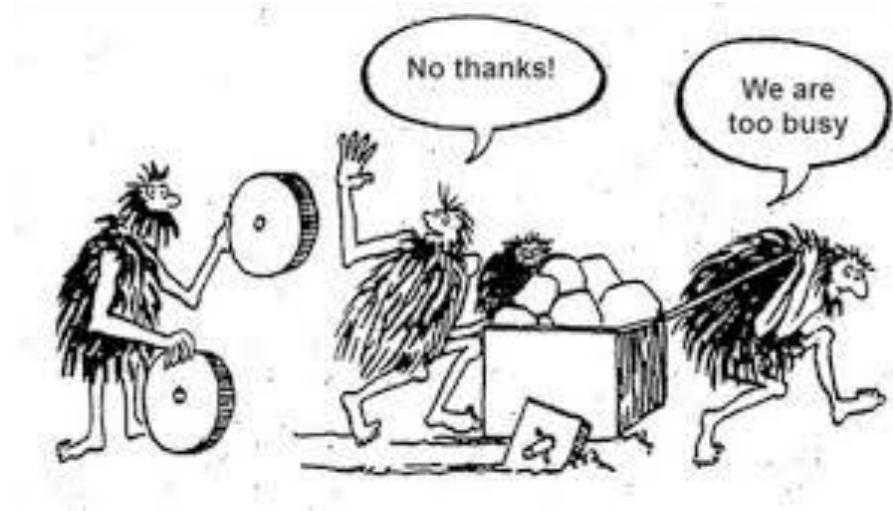
Why should I do the Exercises?

- 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

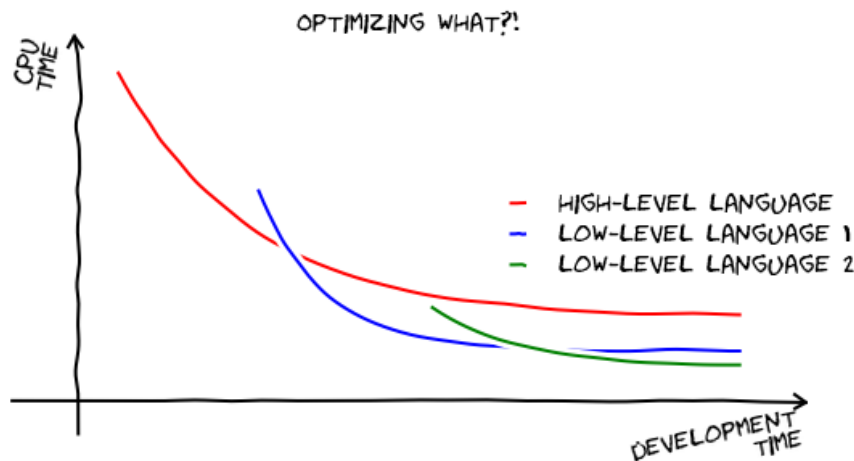
- S6 Pytorch Clasification
- S7 Semantic Segmentation
- S8 Recurrent NN

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



Simple spectral analysis

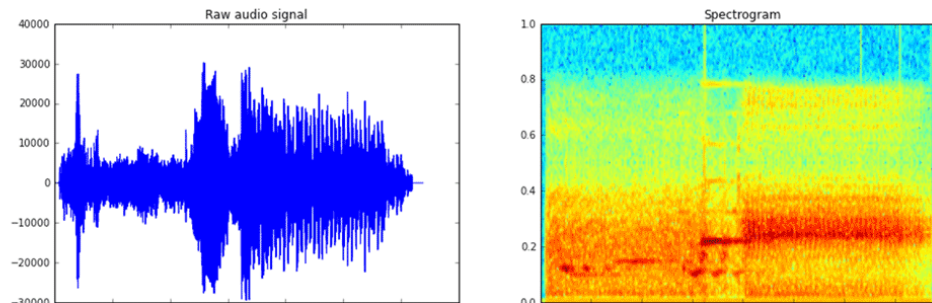
An illustration of the [Discrete Fourier Transform](#)

$$X_k = \sum_{n=0}^{N-1} x_n \exp\left(\frac{-2\pi i}{N} kn\right) \quad k = 0, \dots, N-1$$

```
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 spectrogram routine:

```
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.spectrogram(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 1000² 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 1000² 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/)
(<https://dvl.in.tum.de/teaching/submission/>)



Dynamic Vision and Learning Group

[Team](#) [Research](#) [Publications](#) [Software](#) [Teaching](#) [Contact](#)

Exercise Submission for I2DL

Login

Login

Password

Login with registered account

Register

Matriculation number

12345678

Register account

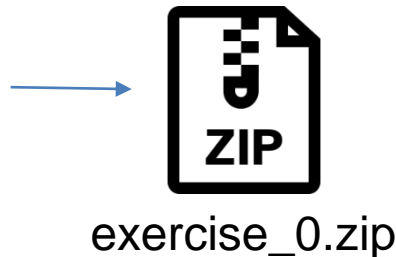
How to submit exercises

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

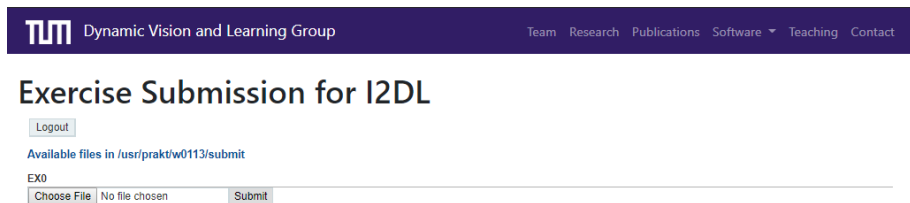
- Submit code + model with
 - Upload code + model for respective exercise by zipping them (no rar, gzip)
 - Note: You will submit your whole code folder as well as your trained models

exercise_code	11/6/2019 5:52 PM	File folder	
images	10/21/2019 10:40 ...	File folder	
models	11/6/2019 5:54 PM	File folder	
1_introduction.ipynb	11/6/2019 6:01 PM	IPYNB File	13 KB
2_data_preparation-optional.ipynb	10/21/2019 10:40 ...	IPYNB File	181 KB
3_colab_introduction.ipynb	10/21/2019 10:40 ...	IPYNB File	4 KB
create_submission.sh	11/6/2019 2:20 PM	SH Source File	1 KB
download_datasets_linux.sh	10/21/2019 10:40 ...	SH Source File	1 KB
download_datasets_mac.sh	11/6/2019 3:51 PM	SH Source File	1 KB



How to submit exercises

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



The screenshot shows the submission interface for the I2DL exercise. At the top is a dark blue navigation bar with the TUM logo and the text 'Dynamic Vision and Learning Group'. To the right of the logo are links for 'Team', 'Research', 'Publications', 'Software' (with a dropdown arrow), 'Teaching', and 'Contact'. Below the navigation bar, the page title 'Exercise Submission for I2DL' is displayed. Underneath the title is a 'Logout' button. A line of text indicates the available file path: 'Available files in /usr/prakt/w0113/submit'. Below this, the exercise identifier 'EX0' is shown. At the bottom, there is a file upload section with a 'Choose File' button, a text box containing 'No file chosen', and a 'Submit' button.

TUM Dynamic Vision and Learning Group

Team Research Publications Software ▾ Teaching Contact

Exercise Submission for I2DL

Logout

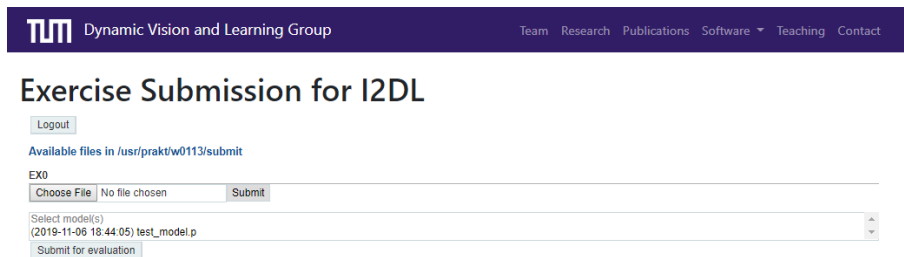
Available files in /usr/prakt/w0113/submit

EX0

Choose File No file chosen Submit

How to submit exercises

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

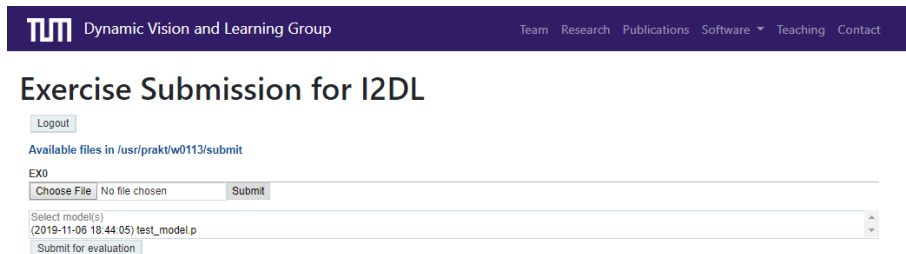


The screenshot shows the 'Exercise Submission for I2DL' webpage. At the top is a dark blue header with the TUM logo and navigation links: Team, Research, Publications, Software, Teaching, and Contact. Below the header, the page title 'Exercise Submission for I2DL' is displayed. A 'Logout' button is visible. The page indicates 'Available files in /usr/prakt/w0113/submit'. Under the 'EX0' section, there is a file upload area with a 'Choose File' button, the text 'No file chosen', and a 'Submit' button. Below this is a dropdown menu for 'Select model(s)' with the selected option '(2019-11-06 18:44:05) test_model.p'. At the bottom of the form is a 'Submit for evaluation' button.

- Select the model you want to evaluate

How to submit exercises

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



The screenshot shows the 'Exercise Submission for I2DL' webpage. At the top is a dark blue header with the TUM logo and 'Dynamic Vision and Learning Group' on the left, and navigation links for 'Team', 'Research', 'Publications', 'Software', 'Teaching', and 'Contact' on the right. Below the header, the title 'Exercise Submission for I2DL' is displayed. A 'Logout' button is on the left. The text 'Available files in /usr/prakt/w0113/submit' is shown. Below this, the exercise name 'EX0' is listed. There is a file upload section with a 'Choose File' button, the text 'No file chosen', and a 'Submit' button. A dropdown menu for 'Select model(s)' shows '(2019-11-06 18:44:05) test_model.p'. At the bottom, there is a 'Submit for evaluation' button.

- 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

```
"""Test Class."""
# pylint: disable=invalid-name

class TestClass(object):
    def __init__(self):
        """
        # TODO:
        # Write some message for yourself
        #
        # Hint: Sometimes we provide some guidelines here.
        """
        self.text = "Replace me..."

        """
        #
        # END OF YOUR CODE
        """

    def write(self):
        print(self.text)

class TestModel(object):
    def return_score(self):
        """
        # TODO:
        # Give yourself a score
        #
        # Hint: You pass if you are better than 50
        """
        score = 42

        """
        #
        # END OF YOUR CODE
        """

        return score
```

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

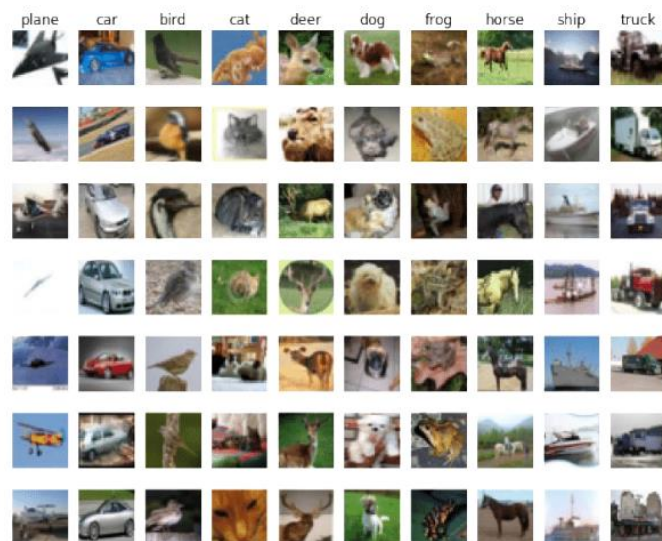
- Dummy exercise – not graded
- Just for getting familiar with the system and for setting up the dev-environment
- Available on moodle

Exercise 1

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

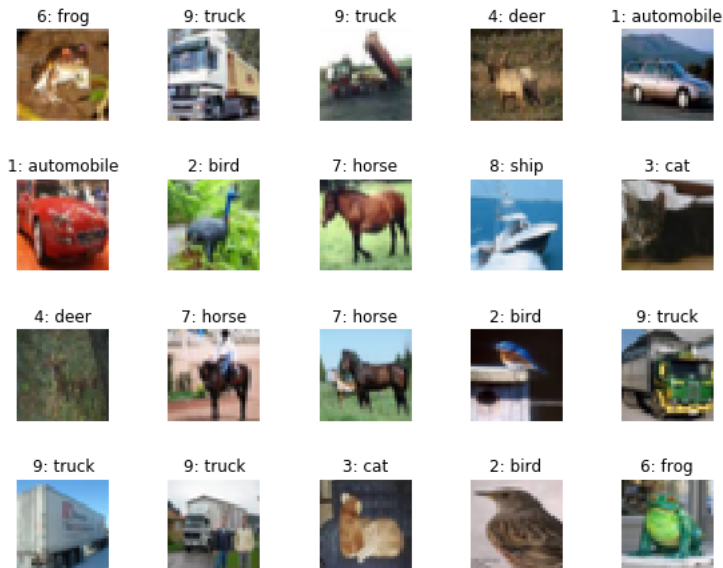
Exercise 1

- 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



Linear
Classifier



?

Exercise 1

- Two layer NN



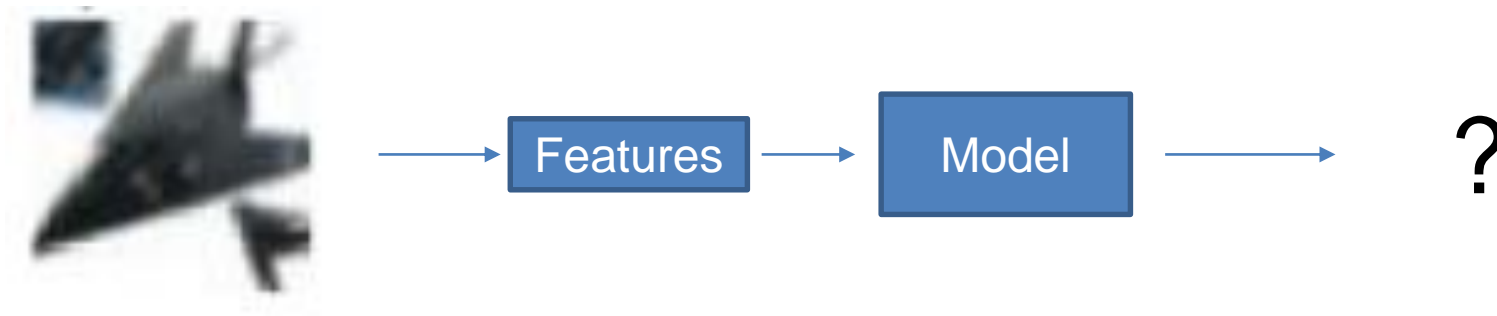
2 Layer
NN



?

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 tutorials: <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?