



FAU

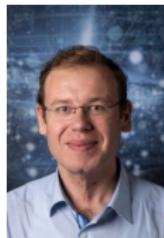
FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

Numpy Tutorial

K. Breininger, F. Denzinger, F. Thamm, Z. Yang, N. Maul, F. Meister, C. Liu, S. Jaganathan, L. Folle,
M. Vornehm, A. Popp, B. Geissler, S. Mehltretter, N. Patel, V. Bacher, K. Fischer
Pattern Recognition Lab, Friedrich-Alexander University of Erlangen-Nürnberg
October 31, 2020



Who are we? - Lab Members



Andreas
Maier



Katharina
Breininger



Felix
Denzinger



Noah
Maul



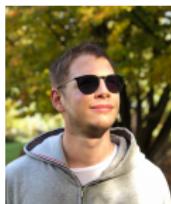
Lukas
Folle



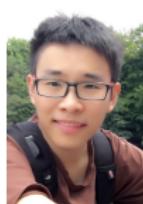
Chang
Liu



Vincent
Christlein



Florian
Thamm



Zijin
Yang



Srikrishna
Jaganathan



Felix
Meister

Who are we? - Student Members



Benjamin
Geissler



Stephanie
Mehltretter



Nupurkumari
Patel



Klaus
Fischer



Marc
Vornehm



Antonia
Popp



Valentin
Bacher



FAU

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

Organisation



Contact us ...

- in StudOn
- via the tutors mailing list: cs5-deep-tutors@lists.fau.de
- in MS Teams

Important: Don't hesitate to ask questions/give comments!

Covid-19 - Online teaching

- Exercise will be online until further notice
- Guide in StudOn → **READ IT!**
- We will use MS Teams (caution: link in German)
 - Team activation in IDM required!
 - "General" channel for general questions and comments
 - "Private" channel for each exercise day
 - Direct support during exercise hours can be requested in resp. channel



Important: Feedback and suggestions very welcome!
Even more important: Stay healthy!

Semester plan

- Five exercises:
 0. Python + Numpy Recap and Data Generation (optional but recommended)
 1. Fully Connected Networks
 2. CNNs and Optimization
 3. Regularization and Recurrent Neural Networks
 4. Image Classification with PyTorch

Semester plan

- Five exercises:
 0. Python + Numpy Recap and Data Generation (optional but recommended)
 1. Fully Connected Networks
 2. CNNs and Optimization
 3. Regularization and Recurrent Neural Networks
 4. Image Classification with PyTorch
- Materials available in StudOn
- Each exercise takes 2-4 weeks, overlap between assignments
→ start early, submit early
- Exercise 1.-4. are mandatory.

Submission

- Group submission possible - pairs of two
- Personal submission only
- Unit tests must pass
- Explain your code
- Upload your code to StudOn
 - please use the provided script (*dispatch.py*) to prepare your upload

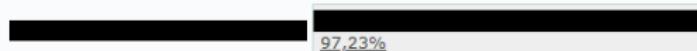
No Plagiarism!

- Plagiarism is strictly forbidden
- We will check that with plagiarism software!

Verteilung - Exercise 4: AlexNet and ResNet in TF / AlexNet and ResNet

90% - 100%	1	#
80% - 90%	2	#
70% - 80%	13	##
60% - 70%	61	=====
50% - 60%	172	#####
40% - 50%	245	#####
30% - 40%	421	#####
20% - 30%	314	#####
10% - 20%	46	====
0% - 10%	0	.

Gruppierte Übereinstimmungen (90% - 100%)





FAU

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

Exercise Setup



First part:

Build a neural network from scratch using test based development

- Implementation task is defined by **description** and **unit tests**
- No skeletons
- Every function and structure is built as a layer
 - As a class in its own file
 - Mandatory functions `__init__()`, `forward()`, `backward()`
- Unit tests help to expose bugs and errors
 - Tested and debugged with python3

Second part:

Build some common neural networks with PyTorch

- Some functionality provided
- No unit tests





FAU

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

Python Overview



About Python...

- Programming language with good readability
- Interpreted scripting language
 - Relies on the call of libraries written in lower-level programming languages
 - Basic programming semantics exist but are very inefficient
- Huge amount of libraries for all sorts of applications



About Numpy...

- Essential python package
- Central object: Numpy array
 - Acts like a matrix/vector
 - Enables all sorts of mathematical operations
 - Optimised for speed
- A cheat sheet with handy functions for this exercise can be found in the StudOn group



About Scipy...

- Python package closely linked to numpy
- Provides additional functionality
 - Signal processing
 - Statistical operations





FAU

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

Recommendations



Package Manager (not needed in CIPs)

We recommend **Anaconda** (Windows)

- Open source
- One click installation
- Also installs python
- Easy handling of virtual environments



IDE

We recommend **PyCharm**

- Open source
- Easy package handling
- Debugging possibilities
- Free licenses for professional version for students



One alternative: Visual Studio Code with Live Share
Plugin (allows remote pair programming)

Version Control

We recommend using GitLab!

- Please use the university's gitlab server: <https://gitlab.cs.fau.de/>
- Perfect for co-working
- Compare your code with old versions
- Please use **private projects**! You can add your study partner as additional developer.



FAU

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG
SCHOOL OF ENGINEERING

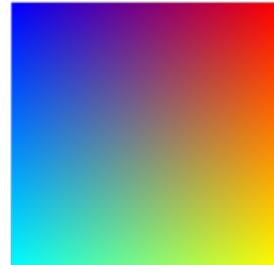
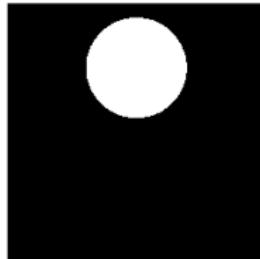
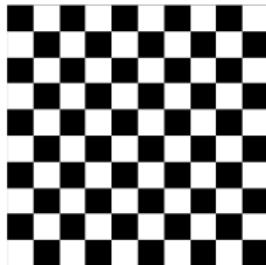
Today's Exercise



1st Task

Use basic numpy functions to create

- A binary checkerboard pattern
- A binary circle
- An RGB color spectrum



2nd Task

Use numpy to implement an image generator class which enables data augmentation.

- The generator yields so called batches (subsets of the training data) in an iterative manner.
- Batch in this context means a set of images, which are returned at once (by calling "next").
- These batches of images must be returned together with their corresponding labels.
- It returns batches until no training samples are left. That is also known as one epoch.

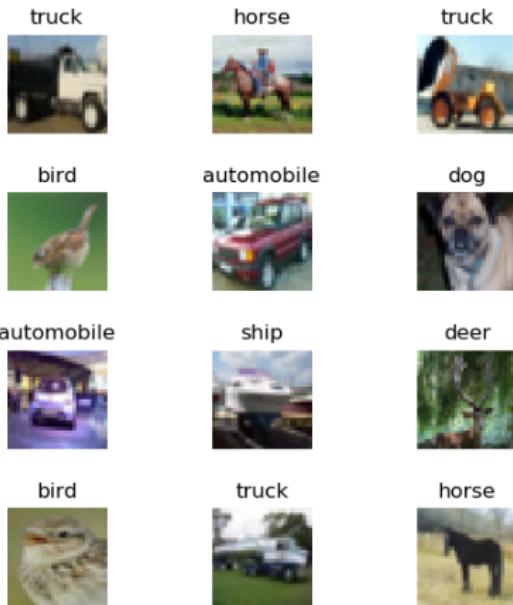


Figure: Example image generator output.

Get Started

- Open the IDE of your choice
- If you want to use PyCharm in the CIP:
type **module load pycharm-community** into the console and open it by
typing **pycharm**
- Follow the instructions of the exercise sheet
- Implement the tasks

Thanks for listening.
Any questions?