Numpy Tutorial

Leonid Mill, Tobias Würfl, Katharina Breininger, Nishant Ravikumar, Sebastian Gündel, Mathis Hoffmann, Florian Thamm, Felix Denzinger

Pattern Recognition Lab, Friedrich-Alexander University of Erlangen-Nürnberg

October 17, 2018

Organisation

Exercise Schedule

Week	Task
15.10.-19.10.	Presentation Numpy Tutorial
22.10.-26.10.	Presentation Exercise 1
29.10.-2.11.	Deadline Numpy Tutorial
5.11.-9.11.	Presentation Exercise 2
12.11.-16.11.	Deadline Exercise 1

Submission

- Group submission possible - pairs of two
- Personal submission only
- Unit tests must pass
- Explain your code

Contact

Don’t mind asking

- During your assigned exercise
- In the studon forum
- Via E-Mail → **cs5-deep-tutors@lists.fau.de**

Cipmap

- Go to cipmap.cs.fau.de/huber
- On the left side click lecturemode - the hand
  → Colored computers represent open requests
- Click **Request Tutor** to open a request
- Click the button again to pull back the request as soon as you get served by a tutor

Numpy Overview

About Python...

- Programming language with good readabilty
- 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

Exercise Setup

First part:

Build a neural network from scratch

- No skeletons
- Every function and structure is built as a layer
  → As own class in its own file
  → Mandatory functions **__init__()**, **forward()**, **backward()**
- We provide unit tests
  → Tested and debugged with python3

Second part:

Build some common neural networks with tensorflow

- Some functionality provided
- No unit tests

TensorFlow

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

Version Control

We recommend using Gitlab!

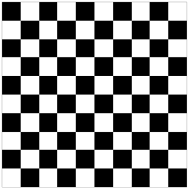
- Please use the university’s gitlab server: https://gitlab.cs.fau.de/
- Perfect for co-working
- Compare your code with old versions
- Please use **private projects**! You can add your study partner as additional developer.

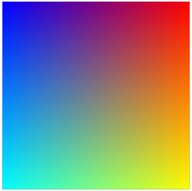
Today’s Exercise

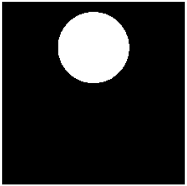
Tasks

Use basic numpy functions to create:

- A binary checkerboard pattern
- A RGB color spectrum
- A binary circle
- Image generator class that enables data augmentation







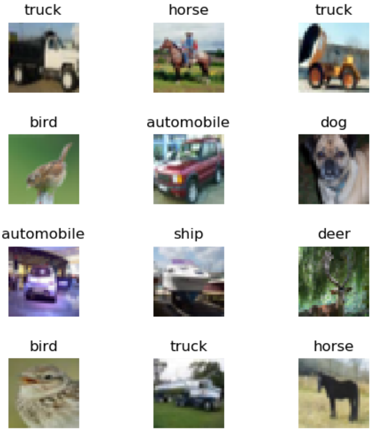


Figure: Example image generator output.

Get Started

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

Thanks for listening.

Any questions?