IANNwTF overview

Leon Schmid

Oct. 2023

Contents

Abstract

This document summarizes the information necessary for participation in the course IANNwTF at UOS in Winter 2023/2024. Please work through it carefully. This document is subject to change, but students will be notified via email regarding any important changes.

1 Course Information

The course *Implementing Artificial Neural Networks with Tensorflow, IANNwTF* is held in Winter-Semester 2023/2024. It represents an introduction to the theory of Deep Learning and its respective Implementation using the TensorFlow library.

1.1 Hybrid format with in-person preference

The course is held in a hybrid form with in-person preference. This means it is technically possible to complete the course by participating online-only (attendance for grading might be required), however, we can not guarantee the same level of support for online-only students. Participation in in-person teaching is strongly recommended for successful participation in the course. However, we will try to support online participation wherever reasonably possible. We do however **not** have the capacity to guarantee these are available, as we have encountered multiple technical issues with this approach in the past iterations of the course. If you can attend in person, we therefore strongly recommend making use of the support structure offered. If you want to attend in an online-only fashion anyway, (1) be aware of this situation, (2) make sure to act proactively, and (3) communicate with the course teaching team if any issues arise.

1.2 Participation

To pass this course, students are required to pass homework submissions. Course grading is based on a midterm exam and a graded final project. The course encourages interaction and active participation, additional points can be awarded for both. There will bean online lecture every week, additionally, a Flipped Classroom and multiple Coding Support / QnA sessions will be available. The course is recommended for Masters or and Bachelors students fulfilling the respective preliminaries.

2 Course Description

Deep Learning fundamentally addresses the creation of certain types of Artificial Neural Networks and how to train them. The course focuses on network archi-

tectures and training fundamentals, which are the foundation of state-of-the-art models in many (and particularly most complex) Machine Learning Challenges.

The course follows four major phases, which partially interleave: First covering the basics of understanding what Deep Learning is, by covering the basics (MLPs, Backpropagation, Optimization, Regularization, Implementation with TensorFlow, Data Handling and Loading). Secondly we study the most common network architecture types, including CNNs, RNNs, AutoEncoders, Generative Models, and Attention-based Networks such as Transformers. Third we cover typical applications (CV, NLP, RL) and give an overview over commonly used tools (Datasets and libraries). Finally in the last (fourth) phase, students work on their own final project in small groups, implementing and summarizing their own (research) projects.

3 Contacts, Teaching Staff

3.1 Teaching Staff

- Instructor: Leon Schmid (lschmid@uos.de)
- Tutors: Mortimer von Chappuis, Tim Niklas Witte
- Staff: Yet unknown

While the course is held under supervision of Prof. Elia Bruni, please regard Leon Schmid as your instructor. The listed tutors will offer Tutorials and support the course in many other ways.

3.2 Contacts

If there are any open questions regarding the course (content, organization, anything else) please contact Leon Schmid via email (lschmid@uos.de). Please refrain from contacting the course supervisors unless this seems necessary to you; they will generally just forward any questions regarding the course to Leon Schmid. Please contact the tutors only regarding questions, which are specific to them, e.g. regarding the coding classes/tutorials they are involved in.

3.3 Communication & Interaction

We will be using the following communication platforms in this course:

- Email: For anything important/official. Please make sure to check your UOS mail account regularly.
- Meetings: Flipped Classroom and Coding Support Sessions will happen live at university (not yet officially confirmed), and streamed live on StudIP if technically possible (check the respective tab in the course!). Live participation is expected (i.e. it is possible some information is only

available in live meetings, however, we actively try to update you also via other channels regarding anything important)

- Lectures: StudIP Courseware System (listed in the StudIP course)
- Discussion: StudIP Forum
- Student-to-Student support: Student Telegram group Student Telegram Group

Please feel invited to actively take part and communicate in the course. Given the advanced and more involved nature of the course, both student-to-student and student-to-lecturer communication will be key to your success in this course, even more so in a hybrid setting.

For any communication, please keep the following in mind:

- There are no stupid questions (!). If something is unclear, please ask away!
- Please stay respectful and helpful to your fellow students and the teaching staff. We expect you to contribute to an inclusive and supportive atmosphere throughout this course!
- Please respect the teaching staff! This includes checking the available resources and reading your email - any question is welcome, especially if you are not sure about a ruling. However we expect you to first check with available resources (this document, Courseware, Emails) before writing an email.

As such, if you have a question about course content:

- Option 1: Ask in the StudIP forum! Fellow students and teaching staff will try to answer your question.
- Option 2: Ask in the Flipped Classroom Sessions! The instructor and/or fellow students will answer your question.
- Option 3: Ask in the Coding Support / Tutorial Sessions! The tutors will answer questions.
- Option 4: Ask Leon via Email or after one of the interactive sessions. They will answer your question.
- Option 5: Ask Leon at their office in 50/402a, they generally have an 'open door policy' and tend to be available there between 10am-8pm. Please knock and be aware the lecturer might be in a meeting at any given time. Please make sure to thoroughly prepare your question and not waste time. You may 'book' a short (up to 10 minutes) meeting by sending a calendar invite to their mail address beforehand.

If you have a question about organizational elements in the course:

1. Step 1: Check whether the question has been answered in this Document.
2. Step 2: (optional): Check in with your fellow students, maybe they can answer your question.
3. Step 3 (optional): Check in with the teaching staff at any Flipped Classroom / QnA / Coding Support
4. Step 3: Just ask the lecturer (mail/in office). Please include [IANNwTF] in your email header.

4 Course and Teaching Statement

The course establishes a shared project between instructors and students to learn more about Deep Learning. As such it is the explicit goal to create a space that enables learning, and where participants mutually support each other in this endeavor. As such we can establish a few key pillars for this lecture:

- **Responsibility**. As instructors and participants, we are both responsible for the success of this course - albeit in different roles. Instructors are responsible for creating the respective spaces and content for participants to learn. Participants are responsible for actively using these. We have a shared responsibility not only for learning but especially for supporting our fellow students (and allowing them to help us respectively!). We also believe in personal responsibility for students: We have a 'no questions asked' 10 free late days for homework submissions rule. The midterm exam is a held as an open book (including non-interactive online contents!) exam. The final project allows for extremely individual projects. We support all of these!
- **Mutual Support and Active Participation** Students are expected to **actively participate, not only consume** in this course. Between the Forum, Lecture Summary Documents, cooperation on homework, flipped Classroom Session and final projects this course offers broad possibilities for peer to peer support. Take responsibility and use them! This course actively encourages peer to peer support by offering multiple platforms for it and additional Bonus Points for students making use of these.
- **Feedback Loops** Multiple feedback Loops are built into this lecture: Student to student feedback loops for homework submissions, lecture questions, in Flipped Classroom and the final project creation exist. The weekly homework submission form includes an optional feedback form. Students are asked to make active use of these tools, to (1) learn how they can improve from the feedback they get, (2) help fellow students improve via feedback and (3) using the feedback form to help shape the lecture into what works best for them!

- **Understanding is key**. There are two sides to this pillar: This lecture will be rather **intensive**, similar to what is known from Computer Science and Math coursees. The topic at hand is complicated, with complex pieces building up on each other. This includes integrating theoretical with engineering perspectives - the topic is hard to cover and understand and we are fully aware of this. Crucially we will try to generally not only learn about these topics, but try to understand why they work, and not only understand such problems in theory, but actually get them to work in practice. As such students are expected to put in the effort to understand and work with complex topics. At the same time, the teaching staff will try to make this **as accessible** for everyone **as possible**: Flipped Classroom Sessions, Coding Support Sessions, Forum and Lecture Summary Documents are all tools to help students in this process, and we generally try to make the lecture as explanatory as possible.

- **Accessibility**. Knowledge is **for everyone**. Many of us have to deal with a variety of situations, different between participants - students participating in Osnabruck, digital-only students, participants with other responsibilities between work, family, and care-work, and participants with different backgrounds and personal situations, including neurodiverse participants. This course is created to be inclusive and enables hybrid participation, which hopefully supports participation with a variety of different backgrounds and situations. Crucially we very deliberately have created the course schedule such that we do not further increase participant stress levels at the end of the semester and have options in place for students struggling psychologically with classical examinations.

  We are fully aware, that there are probably plenty of difficult participant situations, which we have not sufficiently optimized for; If you feel like you might run into struggles with this course due to anything - your personal situation, psychological issues, discrimination or anything else, please feel invited to contact the instructor. Wherever possible, we will create individual solutions!

- **Flexibility and clear communication**. If there is a better way to handle something in the course, we will try to adapt it! Between an unknown number of participants, the partially hybrid setting, and the comparatively experimental course structure, we had to plan for many contingencies. Based on feedback and observed participation we will try to optimize this course on the run, as long as this can be done without creating chaos for participants. Again: If you feel something should change, give feedback, and we will try to accommodate it! If anything changes with the course, we will try to communicate this as early as possible. If you run into issues we also kindly ask you to communicate as early and as clearly as possible. Again - we are committed to making things possible wherever possible, but this requires your active participation!

5 Course Plan

This section outlines the planned structure of the course. Please understand we might rework parts of the curriculum, as such some details might change!

5.1 Course Contents

Week 1	Introduction to Neural Networks
Week 2	Training NNs I: Gradient Descent & Backpropagation
Week 3	Tensorflow Introduction
Week 4	Training NNs II: Optimization
Week 5	Architectures I: CNNs (and Applications 0: CV)
Week 6	Training NNs III: Regularization, Normalization, Hyperparameter Optimization
Week 7	Architectures II: RNNs
Week 8	Architectures III: Autoencoders
Week 9	Architectures IV: Generative Models
Week 10	Applications I: NLP
Week 11	Architectures V: Attention
Week 12	Applications II: Deep Reinforcement Learning
Week 13	PyTorch & other libraries

5.2 Course Structure

The weekly course structure includes a (120-150 min) lecture, the flipped classroom session, a homework assignment and coding support sessions:

- Lectures:
  Lectures are offered as asynchronous digital lectures via courseware on StudIP. They come in a blog-style mixture of text and video clips.
  - Time: Wednesday (uploaded ± 10am)

  If possible we try to upload content early. Generally you will have the lectures from the last year available early, however please be advised it is subject to change until officially released!

- Flipped Classroom
  A Flipped Classroom Session is offered, which supports understanding of lecture contents and improving implementation skills by asking questions, working on small training assignments and create snippets for the course summary. Generally this means small coding or content puzzles will be offered, solved by students (optionally in small groups and with feedback by the organizer), and respective solutions subsequently discussed. While participation is not strictly necessary, we strongly recommend participation as preparation of homework and review of content.
  - Time: Monday (2.15pm - 3.45pm)

- Coding Support Sessions / QnA
  Coding Support Sessions are also called QnA sessions and create a hybrid (digital and live) space for students to work on their assignments and have a tutor available to directly help out with any coding and content questions they might come up with.

  – Wednesday 12.15 - 13.45, 14.15-15.45

  Coding Support Sessions / QnA do not offer any new content! They exist for students to ask questions in small groups and get support while working on their homework submissions. These sessions are shared with the Scientific Programming in Python class!

- Homework Assignments
  For the first part (lecture) of the course, we will have weekly homework assignments covering the contents from both lectures of the respective week. Homework is by each student alone, but you are allowed to share a submission with up to 2 other students (i.e. submitting the same code, but in your respective different repositories). Homework will generally be in the form of an open problem statement ('implement xyz from the lecture material, using tools abc'), so students have to learn to solve the given problems in an end-to-end form.

  – Time: uploaded Wednesday 10 am, deadline Sunday 11.59 (10 days later) (exception: Week 1 due to Student body Lohra Trip)
  – An additional 12 late days are available for each student.

- Support platforms
  The Forum and LaTeX summary documents for every lecture are available for students to facilitate QnA style student-to-student support (teaching staff will of course help out wherever needed!). Actively making use of these is highly recommended and students can obtain Bonus Points for active participation here!

  – Time: Asynchronously, whenever needed by students

6 Participation and Assignments

6.1 Lecture

In each Week there will be a online lecture available. You can find them on the Courseware Tab on StudIP. Lectures generally have the goal of introducing and explaining content step by step. As they are available online, students can work with them completely asynchronously, however, we **strongly** advise working through each lecture at least until the respective following Flipped Classroom, such that students can actively make use of it.

- Work through the digital lecture contents

- Summarize what you have learned, explicitly write down any remaining open questions
- This can be achieved either alone or with a group
- If you are interested in having a shared public viewing for the online contexts, please let the team know in the first Flipped Classroom. If a sufficient number of students is interested in this, we will try to organize two weekly times/rooms for shared viewing of lecture contents
- If anything is unclear, write a question in the respective StudIP Forum Tab
- Check out whether you could help out any other students with their respective questions
- (Optional) Contribute a question and answer (QnA) or subtopic summary to the Lecture Summary Document (and be awarded a Bonus Point)
- (Optional) Contribute any open questions via email or in person for the upcoming Flipped Classroom

6.2 Flipped Classroom

Once per week we have a flipped Classroom meeting[1]. The goal is to review critical lecture content and work toward practical implementation. While we obviously do not check attendance during these meetings, they are highly recommended for successful course participation and students are generally expected to attend. This is especially crucial for student-teaching communication.

- We focus on a student-centered setting here, i.e. wherever possible students get the chance to solve problems themselves, and then discuss their solutions with teaching staff
- Choice of topics is also student-driven: Based on lecture questions forwarded by students, discussing some exemplary pieces of (outstanding) homework submissions, problems we observe multiple groups struggled with implementation, topics students propose
- (Optional) Consider contributing to the flipped Classroom by either submitting outstanding Homework or submitting short (2-10 min) contributions, which can also be awarded a bonus point each.
- Work on small Assignments that deepen your understanding of lecture content in the Flipped Classroom

[1] This is similar to what is often called *Übung* in German university system, but much more student-focused. We want to have a full flipped Classroom, but this is only possible with respective student participation, such that we commonly end up in some middle-ground between *Übung* and flipped Classroom

- Contextualize your understanding of the lecture content in the flipped Classroom, being shown how the topics can be applied and work together with earlier topics from the lecture
- Prepare and discuss homework assignments

6.3 Homework Assignments

The goal of our homework assignments is generally to (1) deepen students' understanding of lecture contents, (2) learn about the engineering problems and efficiency needed for successful Deep Learning applications, and (3) help you get some hands-on experience, which you can also practically show on your repository.

- Homework assignments are given on a weekly basis
- Homework is submitted by each student. Up to three students are able to submit a shared submission at any given week, but are not required to continue this for any subsequent assignments
- Assignments will be uploaded in the homework folder on studIP
- Some assignments are partially prepared in the Flipped Classroom Meetings
- Assignments generally task you with open problem statements
- **Collaboration between students is explicitly encouraged** (make sure you understand everything, not just steal other peoples solutions though!)
- Are submitted via the respective Homework Submission & Feedback Form 10 days later
- Each student has to submit their homework themselves!
- Each student has a total of 12 'late days' - days you can use without further explanation to submit homework late. Please use these responsibly, as they are meant to cover any (small) unforseen events like colds, unexpected appointments, etc. (if something larger, costing you a week at once comes up, we will of course allow additional late days)
- Outstanding Homework submissions can be marked as such on the submission form. Outstanding Homework has to be submitted (1) on-time[2], (2) solved completely, (3) with high-quality code, and (4) requires joining the following two flipped Classroom meetings, where the student may be asked to showcase their solution

[2] no late-days may be used!

6.4 Project

Details about the project work will be discussed in the course. Generally students are to explore DL literature, and finally come up with project ideas, which are then tackled by teams of up to three students in the semester break. More information will be available at the end of the semester.

7 Grading

7.1 Grading Scale

Grading is based on the total percentage of points achieved during the course.

Grade	min. Percentage	min. Points
1.0	95%	190
1.3	90%	180
1.7	85%	170
2.0	80%	160
2.3	75%	150
2.6	70%	140
3.0	65%	130
3.3	60%	120
3.7	55%	110
4.0	50%	100

- Notice: 1 point = 0.5%

Points can be achieved in the midterm exam and the final project; Optional Bonus points can be achieved by actively contributing to the lecture. Students need to achieve at least a passing grade (i.e. passing equivalent percentage) in both their final project and examination to pass the course itself. Students are required to hand in all homework to pass the course. Each student can use a total of 10 'late days' for homework submissions.

7.2 Midterm Exam

The Midterm Exam awards 35% (up to 70 points) of the total grade. The exam will take place around the winter/christmas holidays. The exam will be held as an in-person digital open-book exam via StudIP in the virtUOS digital testing rooms. Questions will reflect this setting and generally require applying or discussing lecture content rather than just recalling it. Students who struggle with the typical examination setting (e.g. due to psychological issues) are invited to contact the lecturer and can be offered an alternative form of examination [3]. The deadline for applying to this alternative midterm examination is November 29th. Participants who can not participate due to

[3] Generally in the form of a written literature review on a subtopic in DL

sickness, or other important reasons (e.g. COSMOS students), will be offered an alternative midterm examination in the form of oral examinations.

7.3 Final Project

The final project awards 65% (up to 130 points) of the total grade. The grading of the final project is based both on implementation and documentation. The current draft (subject to change until Jan. 30th) can be found here: Draft of Final Project Evaluation Sheet. We will provide more details on the final projects at the end of the semester.

7.4 Bonus Points

Students can obtain up to 10 Bonus points throughout the lecture. These Bonus points are completely optional, specifically, they do not count for the maximum 100% (200 Points) achievable, on which the final grade is based. They however can make up for any points a student otherwise did not achieve. Bonus points are generally awarded for going *above and beyond* the expected participation, specifically for contributing to the success of. Please notice that Bonus Points are generally much more effort per point than the exam and final project. Some examples for which a Bonus Point would be granted include:

- Finding (and proposing a correction) of lecture material (only non-trivial errors, i.e. spelling errors or notation details are generally not enough)
- Answering an (at least somewhat involved) question asked by a fellow student on the forum, adding the Question and answer to the QnA section of the respective LaTeXlecture summary. Notice: You are invited to ask and directly answer any questions that come up outside of the forum to be eligible for the same Bonus Point!
- Exceptional Homework submissions, which can be used as exemplary solutions for fellow students
- Preparing a small input (±5 min) for one of the flipped classroom sessions (e.g. details on a topic from an earlier lecture) and (help) moderate the online room for the respective session (can also be done in small teams of 2-3 students)
- Adding a summary entry for a specific topic from the lecture to the respective LaTeXlecture summary

Bonus points are only awarded after students apply for them via this online form: Form Link

8 Course Preliminaries

As an introduction to Deep Learning, the course does has some limited prerequisites. For successful participation, students should bring the following:

- Coding: The course will require coding in Python and specifically training Neural Networks with TensorFlow. We do not expect any prior experience with DL libraries, but general coding abilities. You should e.g. be able to do the following:
  - Write Python classes and functions
  - Understand (or be able to somewhat quickly learn) what Python context managers are (*with* keyword)
  - Understand how Python handles functions as objects, particularly usage of *lambda* expressions
- Maths: High School level maths and statistics should suffice for most of the course, some elements from University level introductory maths courses might be useful. Examples:
  - Calculate the derivative of $log(x)$
  - For what γ is $\sum_i [\gamma^i]$ convergent / divergent?
  - What are bias and variance of a statistical model?
  - Explain what a probability distribution, pdf, and cdf are
- Some additional prior knowledge of Machine Learning and AI is generally useful (understanding what supervised/unsupervised learning is, how regression works, concepts of MLE, overfitting, etc.), but not necessary!

9 Mental Health & Nondiscrimination Statement

We are fully aware of participants being in different personal situations, backgrounds, and situations. As denoted clearly in our course pillars, we try to support participation for everyone who puts in the work for this course. If you feel like we have not seen and supported your situation sufficiently, please contact the course instructor (lschmid@uos.de)!

If you are subject of any discrimination (based on gender, race, background, psychological situation or anything else) feel invited to contact the instructor - they will try and get you in touch with people who can support you in your career/situation.

UOS offers further structures against discrimination, including:

- Equal Opportunity Office - focused on gender equality
- International Office - for international students

- ASTA Referat fuer auslaendische Studierende Students bureau for foreign students [4]

10 Getting started

- Sign up for the course on StudIP
- (Optional) Create a GitHub/GitLab Project for your homework submissions
- Make yourself familiar with the organizational sheet (this sheet!)
- Make yourself familiar with the course resources (Courseware, Summary Sheets)
- Make yourself familiar or recap the necessary skills (inc. Python/Tensorflow, LaTeX)
- Make sure you understand the semester and weekly course structure

11 List of Important Documents and Links

- Homework Submission & Feedback Form
- Lecture Summary Document
- Draft of Final Project Evaluation Sheet
- Student Telegram Group
- Bonus Point Application Form

12 Further Ressources

- Francois Chollet (main contributor for TensorFlow and lead engineer for keras) created the very useful book Deep Learning with Python
- Lilian Weng created an awesome blog, which has great posts on many topics in Deep Learning: Lil'log

[4] Organized and elected by students