Response to the Student Feedback on CS-E4740 Federated Learning offered during spring 2023

responsible teacher: Alexander Jung (first.last (at) aalto.fi)

February 5, 2024

We express our sincere gratitude for the insightful and constructive student feedback received via the "Webropol" survey implemented by Aalto University. Major modifications of the course, offered again during spring 2024, include

- **Discussion Forum.** While students consider the course discussion forum helpful, some concerns have been raised about the structure of the forum. Therefore, we now organize the discussion forum according to the main course elements (lectures, assignments). We will also discuss the organization of the discussion forum in the first lecture of the course.

- **Grading.** Many students praised the flexibility of choosing between different graded activities. However, some students disliked the possibility to quickly earn many points by completing bonus tasks. We therefore slightly tighten the grading scheme in the new edition. In particular, the grading will be mainly based on assignments, each covering the topics of a corresponding lecture. Besides these nine core assignments, we will also offer two bonus assignments.

- **Coding Assignments.** We will now offer a coding assignment for each lecture (except the first welcome lecture). The coding assignment requires you to solve student tasks by completing code templates provided as Python notebooks. We describe the student tasks in the notebooks as well as in the lecture notes. In particular, the final section of the notes for each lecture (except the first one) contains an overview of the coding assignment.

- **Quizzes.** To test the understanding ot lecture topics as well as the solutions to the corresponding coding assignments, we will use MyCourses quizzes. We will use coherent format for this quizzes, allowing a single attempt and only revealing correct answers after the assignment deadline. At the end of the course, students have the opportunity to organize a review meeting with the teacher. This meeting allows to review the student performance as well as to earn lost points via answering questions from the tester (somewhat like an oral exam).

- **Feedback.** The feedback on the student solutions to the assignments consists of two components: (i) The correct answers for the corresponding MyCourses quiz answers revealed after the assignment deadline. (ii) A reference solution to the coding assignment will be published after the assignment deadline.

In the following, we respond to selected student feedback in a point-by-point manner, grouped into the following topics: **Logistics**, **Assignments**, **Material**, **Project**. We have tried to select the most relevant student feedback to avoid redundancy in our response letter. The complete student feedback reports (suitably anonymized) are available on request.

I. Logistics

Log1 *Maybe present some industrial federated learning models to catch up with the advancement of the technology.*

We will invite speakers from industry to provide first-hands information about current trends in FL industry.

Log2 *I would add more coding examples/assignments.*

We now offer coding assignments for each lecture (except the first lecture on course logistics).

Log3 *in the last third we switched only to jupyter notebooks as assignments, I would suggest combining quizes and notebooks from the start*

We will now use coding assignments in Python notebooks for each lecture.

Log4 *Some kind of feedback for quizzes after the first round. Maybe the feedback could just say if you received less than 50 % of points or more than 50 % of points for the round so you could reflect.*

We will use quizzes to test your understanding of lecture contents and solutions to corresponding coding assignment (one such coding assignment for each lecture). These quizzes are meant to be like mini exams and therefore allow only one attempt. After a quiz closes we will release the correct solutions to quiz questions and a reference solution to the coding assignment.

Log5 *1) Getting points can almost be too easy. Especially since a lot of the exercises in lecture notes gave a lot of points. 2)Programming quizzes are good, but please use automated grading. I think all of the programming tasks could have been graded automatically, which would improve learning as we get instant feedback, but also it would save you a ton of time, as the quiz results were often weeks late. 3) Perhaps some stuff should be mandatory.*

We have tightened the grading which is now mostly based on quizzes that test understanding of lecture contents and solutions to coding assignments. The feedback will be provided immediately after a quiz closes, in the form of correct quiz answers and a reference solution to the coding assignment.

Log6 *The quiz configurations were odd a lot of the times. Sometimes the quiz said I have 2 attempts but no time limit, sometimes it said there is a time limit but nothing about having limited attempts. For the last two quizzes I could not review my submission, as it simply stated "Not permitted". I understand there seems to be a lot of work happening on developing this course, but consistency would be nice.*

We now use a coherent format for the quizzes, which allow only one attempt. These quizzes serve as mini-exams (which also do not allow to revise your solutions after you handed in) which are immediately graded. After the quiz closes, we reveal the correct answers along with a reference solution to the coding assignment.

Log7 *I think that the point balance between the different course completion methods should be stated prior to the course's beginning and that it should be redone a little bit - especially regarding some of the* *lecture note exercises.*

We now indicate the grading scheme clearly on the MyCourses page and also discuss it during the first lecture of the course. The new course edition does not offer any bonus exercises in the lecture notes.

Log8 *Could be good to notify if/when the course book is updated, as some people download it.*

instead of a book-like lecture notes, we now provide more self-contained notes for each lecture separately.

Log9 *Perhaps points are too easy to get. . . . Perhaps there should be a minimum that you must get points from different categories?*

We have tightened the grading scheme which is now entirely based on MyCourses quizzes that test understanding of lecture contents and solutions to coding assignments. There will be no minimum point requirements for individual quizzes. However, to achieve top grades it is necessary to perform well on most quizzes.

Log10 *The first part of the course which covered the basic could be omitted, to reserve more time for a *compulsory* project.*

We have reduced the review of basic ML techniques, which are relevant for FL, to a single lecture "ML Basics". The student project is still optional (for completing the extended 10 credits course variant) as it incurs a significant workload that cannot be accommodated in the basic course variant (5 credit). The basic variant focuses on the theoretic foundations of FL.

Log11 *I would like to have a bit more labs, maybe 1 every 3 weeks or so maybe combining the coding exercises with quizzes during whole semester*

The new course edition will not include any labs and instead more coding assignments as a "hands-on" experience.

Log12 *The erratic quiz schedule . . . was the only down side I can come up with,*

We apologize for the adaptions on short notice during the first course edition and will do our best to avoid any such adaptions in the new course edition.

Log13 *I would also have liked to have more in person lectures.*

The new course edition will offer more on-site (in-person) lectures which will be recorded and made available to students in a timely fashion. Attendance in these lectures is not mandatory as the course can be completely fully remote.

Log14 *I like the idea of having coding assignments, however I was only considering to take the course by doing the quizzes based on the notes, because my time is limited due to my job. I think this should have been thought before the course starts.*

We now closely monitor the students workload by requiring them to report the estimated workload at the end of each quiz, which covers a lecture and coding assignment. The basic variant of the course amounts to 5 credits which translates roughly into 130 hours of work. The extended variant, including a student project, amounts to 10 credits.

Log15 *Setting for MyCourses quizzes were mixed: sometimes there were 2 attempts+2h limit, sometimes 1 attempt+2h limit or 2 attempts+no time limit. The grading time is a way too long. It took almost 2 weeks to get grading for notebook*

We now use a coherent format for the quizzes: allowing one single attempt with a maximum duration of two hours. This quizzes are meant as mini-exams (which also do not allow another attempt after you handed in the exam paper :-). After the quiz closes, we will immediately release the correct answers and a reference solution to the corresponding coding assignment.

II. Assignments

Ass1 *The coding tasks should be formatted in a more structured way. A lot of time went to just figuring out what is being asked in the tasks.*

We have put more effort into preparing Python notebooks including coding assignments. Moreover, we now also discuss the coding assignments at the end of the notes for each lecture.

Ass2 *Somehow I would like to see the course keep the many different options of earning points but also make them a) more comparable but also b) make sure we learn the key things.*

We have now reduced the diversity of possibilities for earning points. In particular, points can only be earned by completing MyCourses quizzes. However, we do not set minimum requirements for individual quizzes. These quizzes test your understanding lecture contents and coding assignments. The lecture notes now include a listing of learning goals and an overview discussion of the coding assignment. We now also use a coherent format for coding assignments using Python notebooks that include the detailed description of student tasks.

Ass3 *Some exercises needed a bit more instructions*

We now provide more detailed instructions for the coding assignments in the corresponding Python notebooks. Moreover, we now also provide an overview of the coding assignment in the notes for the corresponding lecture.

Ass4 . . . *For example, FL flavours notebook: task was unclear and we have to check slack to really understand what was expected in the task. Notebook consisted of long pieces of code without proper/ gradual introduction.*

see our response to [Ass3].

Ass5 *The course was a bit all over the place with the arrangements often changing so I had to constantly check the messages to know what is going on.*

We now use the GitHub repository click me as the main location for course material and information. Moreover, we will maximally avoid any changes of course schedule.

III. Material

Mat1 *Script structure should be improved to be consistent with Prof. Jung's mathematical approach. By that I mean to change the style of and structure of script to be more like a math book (probably that's done* *in his machine learning book already, I don't know).*

We now include more mathematical analysis in the lecture notes. The format of the lecture notes has been changed from a textbook style more towards hand-outs that can be published separately for each lecture.

Mat2 *More in-depth applications already in the basic version*

We have expanded the extend of mathematical analysis in the lecture notes which now contain some propositions (theorems) along with proof sketches.

Mat3 *The lecture still seems quite new, the lecture notes still need completion and development. I guess the topic and the course are still new. The math somehow feels too much, it would be better if more intuition are provided. For example, the distinction between different algorithms is not very clear. Probably there needs a proper summary somewhere somehow.*

We have added more narrative text to the lecture notes to better motivate and develop intuition for the mathematical concepts used to study and design FL systems. The lectures will also include more explanations and illustrations to better support the learning of important concepts.

Mat4 *I think the course overlap a bit too much with the course basic in machine learning.*

We have trimmed the review of basic ML techniques which is now covered in a single lecture.

Mat5 *There was too much mathematics formulat that I don't understand how to convert them to python language.*

We have added more discussion and explanation of mathematical concepts in the lecture notes. The notes now also contain brief discussion of the corresponding coding assignments.

Mat6 *Also it would be nice to implement the gradient based methods with some autograd library like PyTorch.*

A main goal of the course is to equip students with a firm understanding of gradient based methods including the computation of gradients for linear models. Having a firm grip on gradient descent is instrumental for the analysis and implementation of some widely used FL algorithms.

Mat7 *Course is not in depth enough*

We have tried to increase the depth of discussion and analysis of the core mathematical concepts such as empirical graphs and generalized total variation minimization. The lecture notes now also include few rigorous statements in the form of propositions along with their proof (sketches).

Mat8 *The exercise questions in the lecture notes were sometimes too vague or hard to understand.*

The new lecture notes do not contain any open exercises as bonus tasks.

Mat9 *FL lecture notes were updated continuously during the course, which is very annoying as I have to re-download pdf every time it is edited, so I won't miss anything important. Notes should be ready before the course started, so we can just upload it once and not worry that there might be some edits coming. Otherwise the course is great and I am pretty sure next edition will be fantastic :)*

We now use a different format for the lecture notes. The notes will be more like (fancy) hand-outs that will be released before the corresponding lecture.

Mat10 *I was extremely disappointed by the course, considering I was excited about the subject going in. I would say the entire first half of the course was not about Federated Learning. I understand you need to consider people of all backgrounds for this, but is it really fair to offer the course as a Master level course then? It felt like a surface level introduction to something most people already knew. In general. the lectures about actual federated learning were very few (considering the last lectures were about ethics, which I still found interesting) and I felt like basically one concept and algorithm was presented.*

We have significantly trimmed the review of basic ML concepts which are now mainly covered in a single lecture “ML Basics”. There will be also more discussion about many widely-used FL algorithm arise from a single underlying design principle, i.e., generalized total variation over empirical graphs.

Mat11 *I could not for the life of me formulate a GTVMinimization problem in code. How can I solve an argmin problem in python? Likewise I feel like there were two layers of difficulty in this course, first is the main part and most of the exercises, but then the second appears out of nowhere and we go really deep into the math with no materials but just live on the lecture. I feel like for these parts more explanations would have been needed. And then in the exercises how to then take this into code.*

We have put some effort into making the different course components more coherent. The notes and lectures now provide more explanation and discussion of mathematical concepts such as GTVMinimization and gradient based methods to solve it (i.e., “do the argmin”). There will also be more discussion of how mathematical concepts, such as gradient steps, can be implemented using the programming language Python.

IV. Project

Pro1 *I would like to learn about the popular tools and frameworks commonly used in the industry for federated learning. Additionally, I would be interested in analyzing and discussing more practical applications and real-world case studies related to this field.*

We will now include guest talks from industry. The basic variant of our course focuses on the mathematical theory of FL problems and algorithms. However, the FL project of the extended variant allows to study any FL applications or business-case you wish.

Pro2 *Maybe more clarity about project work schedules in the beginning of the course would have been helpful*

The schedule for the student project is already now fixed and indicated here.

Pro3 *I feel like the course would have benefited from more practical examples of federated learning applications or case studies.*

see our response for [Pro1].