

ARHU299



ARHU299 Syllabus

Syllabus for ARHU299: Machine Learning in Language and Art at the University of Maryland, College Park.

Last updated January 24, 2024.

Instructor: Omar Agha (oagha@umd.edu)

TA: Utku Türk (utkurturk@umd.edu)

Office hours:

Omar Agha: Monday 3-4pm in MMH 3416A (outer door is locked; email Omar to be let in)

Utku Türk: Wednesday 4-5pm in MMH 1407C (outer door is locked; email Utku to be let in)

Course Summary [🔗](#)

In this course, you will learn how machine learning models read and write text, classify images, and more. This requires understanding some basic elements of linear algebra (vectors and matrices), calculus (the chain rule for derivatives), Python programming, and basic statistics. If you have not seen any of this before, you will learn the relevant tools along the way, in the context of solving applied problems, and with the aid of software that performs complex calculations for you.

This is a fast-paced class. There will be homework every week, and frequent in-class assignments. You will need to make a serious effort to learn about difficult technical topics. I assume you are here because you want to learn this material, and I will do everything I can to help you succeed.

We will work up from simple models to more complex ones. We will start with the simplest neural network architecture, the perceptron, and apply it to text classification tasks. After that, we will study deeper networks (multi layer perceptrons) that can be used for language modeling (predicting the next word).

By the end of the course, we will cover the basics of transformer models and convolutional neural networks. Transformers are the building blocks of language models like GPT, and convolutional NNs are used for classifying and generating images. Depending on the speed of your progress, we might also have time to discuss more advanced neural network architectures like latent diffusion models, which are behind powerful text-to-image generators like Stable Diffusion and Midjourney.

How Deep Will We Go?

Engineers and scientists who work on these topics usually do years of intense study in math and computer science before reaching proficiency. So it's not realistic for us to go *that* deep in one semester.

Luckily, building simple text and image models (and maybe music generators) does not require as much expertise, only some Python skills and enough curiosity to figure things out on one's own.

Math Skills

You will learn how to multiply matrices, use probability, and understand simple equations and graphs involving functions like \sin , \cos , and \log . You will see how these concepts are actually applied in real-world problems.

Machine learning models use calculus (derivatives) under the hood, and you will learn the basics of how this works. But for the most part, we will rely on software to do those calculations for us.

Programming Skills

You will learn how to use the Python programming language, along with some Python libraries. The most important libraries are PyTorch and Numpy, for matrix math, and Pandas, for data processing. We will also use Matplotlib, Seaborn, and Plotly for plotting, and Pillow for image processing.

The basic Python concepts you will learn are functions, classes, lists, tuples, dictionaries, `for`-loops, list comprehensions, and `lambda` expressions.

If you want to get a head start on your Python journey, google all the unfamiliar terms in this section and read up.

Resources

There is no textbook, since I haven't found one that fits the audience of this course. I will post optional resources (books, articles, and videos) in the lecture notes, and under the [Resources](#) section of the course website.

Lecture Notes

I will post lecture notes within 48 hours after the relevant lecture. Lecture notes will usually not be available before class, so you must pay attention and take your own notes during class sessions.

All lecture notes will be posted under [Lecture Notes](#). I will also post code in the form of Python Notebooks on the website. You can open a copy in Google Colab to study and play with the code.

Class Policies

My Expectations

I will assume that you are here because you want to learn. This applies to all policies and requirements: I assume that you will come to class and complete the homework because you want to learn.

I will do my best to not waste your time, and to not impose any requirements on you that are unnecessary for your educational goals.

Attendance

Attendance is not directly graded. However, there will be frequent in-class assignments, and participation is a part of your grade. Also, missing class will make it hard to keep up. For this reason, poor attendance will generally lower your grade and prevent you from learning effectively.

Homework

In this class you will learn math, programming, and critical thinking skills by practicing them. Learning this material without practice is not possible.

I find that accepting late homework usually causes more students to fall behind, ultimately harming their grades and progress, even if it feels generous in the short term. So, **late homework will not be accepted** unless I have already granted you an extension.

We will mostly be coding in Google Colab. If you want to use another editor you can, but your submitted code must be in `.ipynb` format and it must run correctly in Colab. (So, make sure to upload your file in Colab and test it there.)

Extensions and Excused Absences

If you need an extension on any assignment, contact me **before noon on the due date**, and verify that I approve. (Better to reach out earlier than that if possible.)

If you need to be excused from class, contact me **before that class starts**.

If you are excused from class, I will let you know if there are any in-class assignments that you need to make up.

Plagiarism

[The UMD Code of Academic Integrity](#) defines plagiarism as “representing the words or ideas of another as one’s own in any academic course, exercise or research”. If you copy text or code from any source, **you must treat it as a quotation and provide the source**. (You can either link to the source or provide a full bibliographic citation.)

Plagiarism on any assignment will result in a zero. Repeat plagiarists will be reported to the Office of Student Conduct.

Copying work from another student is plagiarism, but group work is encouraged. All submitted work must be your own.

ChatGPT/Generative “AI”

Generative “AI” tools like ChatGPT or Bing produce text in a random process by sampling from the most likely next words.

They can be useful in certain contexts. However, the randomly generated text is often biased and false in subtle ways, and plagiarized from other sources. (This is because the model’s next-word-probabilities are derived from documents scraped from the internet.)

Copying and pasting the output of generative AI on any assignment will be considered plagiarism. (See the above definition.)

If you choose to use generative AI to help with solving problems, that’s ok, but you must ensure that all submitted work is your own, and you are responsible for any errors. (Using ChatGPT or other software to edit your prose is also ok, especially if you are not a native speaker of English.)

Grades

Distribution

- 50% homework
- 20% quizzes
- 10% participation (in person and on Discord)
- 20% project proposal

Project Proposal

By the last day of class, please submit a 2-3 page project proposal. This can be for a product that you would like to create, or for a research project. In the proposal, please discuss in detail

1. how you would use the technologies we cover in class, (5%)
2. what you think the most challenging components would be, (5%)
3. what the likely social impact of the project would be. (5%)

You don’t need to turn in any working code. (You are welcome to do so, and you won’t be graded on the code, only the proposal.)

Important Dates

Before Mar 14, please schedule a meeting with me to talk through project ideas.

- **Mar 14:** Turn in one paragraph on your project proposal. If you have multiple ideas, do a short paragraph for each.
- **Apr 18:** Turn in a half page, more fleshed out version. Start to include some details about technologies, challenges, and/or social impact. (Worth 5% out of the full 20%.)
- **May 13:** Final due date for project proposal.

How to Succeed in This Class

Asking For Help

Your success is very important to me. I (or the TA) will answer your emails promptly (within 24 hours) and make time to meet with you if you need extra help.

If you need help, please contact me ASAP. Often, students fail because they wait too long to ask for help, and then fall too far behind to catch up. Don't let this happen to you!

Using Resources

If you find yourself struggling to understand something, try re-reading the lecture notes or re-watching the assigned videos. Ideas often take multiple tries to click.

I will link to helpful videos and articles in my lecture notes. Don't forget to use them!

Group Work

Group work is highly encouraged. Please try to find a study group early on, and try to meet up for each homework. I'll make a Discord chat for the class to help you all collaborate. As a reminder, copying code or prose directly from other students is plagiarism, so please make sure that all submitted work is your own.

Food, Sleep, and Exercise

Taking care of yourself is the foundation for all success in life. Make sure you come to class well-rested, nourished, and ready to learn. (If you need to eat or drink beverages in class, that's fine with me.)

Mental Health

If mental health challenges are keeping you from doing your best, the university provides [Counseling](#) and [Behavioral Health Services](#). The earlier you reach out, the more you can benefit from these resources.

Disability Accommodations

Students with disabilities are very welcome in this class. If you require accommodations, please contact me as soon as possible. For more information about disability accommodations at UMD, please see the [ADS webpage](#).