# CSCI5527: Deep Learning—Models, Computation, and Applications

Spring 2025

## General Information

Over the last few years, deep neural networks (DNNs) have fundamentally transformed the way people think about machine learning and approach practical problems. Successes around DNNs have ranged from traditional AI fields such as computer vision, natural language processing, interactive games, to healthcare and physical sciences—touching each and every corner of theoretical and applied domains. On the other hand, DNNs still largely operate as black boxes, and we only have a very limited understanding as for when and why they work. This course introduces the basic ingredients of DNNs, samples important applications, and throws around open problems. Emphasis is put on thinking from first principles and basic building blocks, as the field is still evolving rapidly and there is nothing there that cannot be changed.

- **Prerequisite:** CSCI5521 or CSCI5523 or equivalent. Maturity in linear algebra, calculus, and basic probability is assumed. Familiarity with Python (esp. numpy, scipy) is necessary to complete the homework assignments and final projects.

- **When & Where:** Tue/Thur 2:30–3:45pm @ Bruininks Hall 22

- **Who:**

  Prof. Ju Sun (Instructor, jusun@umn.edu ) Office hours: 4–6pm Thur (Keller 6-213 or zoom)
  Hengkang Wang (TA, wang9881@umn.edu) Office hours: 10am–12pm Fri (Keller 2-209)
  Wenjie Zhang (TA, zhan7867@umn.edu) Office hours: 4–6pm Tue

- **Public course website:** https://sunju.org/teach/DL-Spring-2025/ All course materials, including course schedule, lecture slides, supplementary reading, homework sets, and project description, will be posted on the course website. We will make announcements in Canvas when we post there; enrolled students are encouraged to check the website on a regular basis.

- **Teaching mode: in-person** We will try to record the lectures and post them in Canvas to help you review the lectures, but that should not be used as excuse to skip the lectures. Our lectures tend to be highly interactive in nature. Also, given the limited resources we get, we cannot guarantee our recording reaches professional quality in any sense.

- **Communication:** Our philosophy is to minimize emailing. **Piazza** is the preferred and most efficient way of communication. Please post all questions and discussions related to the course there, and make them public if possible, instead of sending emails. If you have to use emails for non-technical issues, please begin the subject line with "CSCI5527" so that we can prioritize your emails.

## Tentative Topics

**Lecture sessions** The tentative topics are as follows, and they are subject to change later.

- Course overview
- Neural networks: old and new
- Fundamental belief: universal approximation theorem
- Numerical optimization with math: optimization with gradient descent and beyond
- Numerical optimization without math: auto-differentiation and differential programming
- Working with images: convolutional neural networks
- Working with images: recognition, detection, segmentation
- Working with sequences: recurrent neural networks
- Transformers and large language models
- Working with graphs: graph neural networks & applications
- Learning probability distributions: generative models (GANs, VAE, normalization flow, and diffusion models)
- Learning representation without labels: dictionary learning and autoencoders
- Learning representation without labels: self-supervised learning
- Gaming time: deep reinforcement learning

## Assessment

- Homework $60\%$: $6$ homework sets and $15\%$ each, the top $4$ scores will count toward the final grade
- Course project $40\%$: proposal $(5\%)$ + lightning talk $(10\%)$ + final report $(25\%)$. The project can be survey of a chosen topic not covered in detail in the class, comparison of existing methods, or novel foundational or applied research
- **Bonus points up to** $5\%$: $1\%$ for $3$ good questions and answers in Piazza (marked by the instructor and the TA)
- Final grades on a curve, but will be assigned *generously*

## Homework

You have approximately 14 days (i.e., 2 weeks) to complete each homework (the exact due date will be specified in each assignment). **Late submissions will not be accepted; our flexible 4-out-of-6 HW grading policy captures all exceptions and so please do not ask for exceptions (unless for DRC accommodation or exceptions given to the whole class)**. All submissions *must* be electronic and uploaded via the Canvas system. The written part should be neatly written/scanned or typeset and *must* be submitted as PDF files. For students pursuing research in relevant areas, it is strongly encouraged that you type your solution using LaTeX. Computer programs *must* be submitted in the Python notebook format. Only `Python 3` (no `R`, `Matlab`, etc) will be used and accepted in this course.

Collaboration on homework problems is strongly encouraged, but each student must ensure that the final submission is prepared individually. **Collaborators should be properly acknowledged in the final submission, at the problem level.** The same applies to computer programs. Plagiarism and cheating are not tolerated and are subject to disciplinary action. Please consult the student code of conduct for more information: https://regents.umn.edu/sites/regents.umn.edu/files/2019-09/policy_student_conduct_code.pdf

**About the use of AI tools** You are strongly encouraged to collaborate with AI tools, such as ChatGPT (https://chat.openai.com/) and Claude (https://claude.ai/chats), and Github Copilot (https://github.com/features/copilot) when trying to, e.g., solve homework problems and come up with project ideas. They are becoming our workspace friends. It takes a bit of practice to ask the right and effective questions/prompts to these tools; we highly recommend that you go through this popular free short course **ChatGPT Prompt Engineering for Developers** offered by https://learn.deeplearning.ai/ to get started.

Our catch-it-or-miss-it policy: **If you use any AI tools for your homework problems, you are required to include screenshots of your prompting questions and their answers in your writeup**. The answers provided by such AI tools often contain factual errors and reasoning gaps. **So, if you only submit an AI answer with such bugs for any problem, you will obtain a zero score for that problem**. You obtain the scores only when you find the bugs and also correct them in your own writing. You can also choose not to use any of these AI tools, in which case we will grade based on the efforts you have made.

## Course Project

The course project is to be carried out by teams of 3 or 4 students, and the weight of the project should be proportional to the number of students in the team. All students on the same team will receive the same score for their course project.

The project can be, but not limited to: a survey of literature on a focused topic not covered in class, comparison and improvement of existing methods, novel application of DNN techniques, and novel development of DNN methods and theories. You are encouraged to ask AI resources to generate project ideas for you and even ask them to draft codes for you. Our evaluation will be on the depth you can go: novelty, reasoning, and insights—which current generative AI tools are weak at—that you put into your projects. Use these tools to elevate the level of your project!

## Programming and Computing

Our programming environment will be Python 3. We will use Pytorch as the default deep learning framework, although Tensorflow ($\geq 2.0$) and Jax are also accepted. For small-scale experiments (e.g., typical homework problems), Google Colab (https://colab.research.google.com/; everyone enrolled in the class gets a 3-month subscription to their professional version) will suffice. A local installation of relevant software packages may be a reasonable alternative, although we do not provide technical support for this. For large-scale course projects, we will use the Minnesota Supercomputing Institute (MSI) GPU computing queues based on our class account.

## Recommended References

There is no required textbook. Lecture materials and assigned reading will be the primary resources. Recommended reference books—most of them freely available online—are

- **Dive into Deep Learning** by Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola. Live book; Freely available: https://d2l.ai/ (comprehensive coverage of recent developments and detailed implementations based on NumPy/Tensorflow/Pytorch/MXNet)
- **Understanding Deep Learning** by Simon J.D. Prince. MIT Press, 2023. Freely available: https://udlbook.github.io/udlbook/ (comprehensive coverage of recent developments and detailed implementations)
- **Deep Learning: Foundations and Concepts** by Christopher M. Bishop & Hugh Bishop. Springer, 2024. Freely available: https://www.bishopbook.com/ (comprehensive coverage of recent developments and detailed implementations)
- **Deep Learning** by Ian Goodfellow and Yoshua Bengio and Aaron Courville. MIT Press, 2016. Freely available: https://www.deeplearningbook.org/ (comprehensive coverage of developments by 2016)
- **Neural Networks and Deep Learning** by Charu Aggarwal. Springer, 2018. UMN library online access (login required): Click here. (comprehensive coverage of recent developments)
- **The Deep Learning Revolution** by Terrence J. Sejnowski. MIT Press, 2018. UMN library online access (login required): Click here. (account of historic developments and related fields)
- **Deep Learning with Python** by François Chollet. Online URL: https://livebook.manning.com/book/deep-learning-with-python (hands-on deep learning using Keras with the Tensorflow backend)
- **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems** by Aurélien Géron (2ed). O'Reilly Media, 2019. UMN library online access (login required): click here. (hands-on machine learning, including deep learning, using Scikit-Learn and Keras)
- **Deep Learning for Vision Systems** by Mohamed Elgendy (1ed). Manning Publications, 2020.

## Related Courses

Within UMN

- **Topics in Computational Vision: Deep networks** (Prof. Daniel Kersten, Department of Psychology. Focused on connection with computational neuroscience and vision)
- **Analytical Foundations of Deep Learning** (Prof. Jarvis Haupt, Department of Electrical and Computer Engineering. Focused on mathematical foundations and theories)

- **Theory of Deep Learning** (Prof. Yulong Lu, School of Mathematics. Focused on the recent theoretical developments of deep learning)

Global

- **CS230 Deep Learning** (https://cs230.stanford.edu/, Stanford Computer Science)
- **CS231n: Convolutional Neural Networks for Visual Recognition** (http://cs231n.stanford.edu/, Stanford Computer Science)
- **CS224n: Natural Language Processing with Deep Learning** (http://web.stanford.edu/class/cs224n/, Stanford Computer Science)
- **CS236: Deep Generative Models** (https://deepgenerativemodels.github.io/, Stanford Computer Science)
- **Introduction to Deep Learning** (https://deeplearning.cs.cmu.edu/F20/index.html, CMU, 2020)
- **Advanced deep learning and reinforcement learning** (https://github.com/enggen/DeepMind-Advanced-Deep-Learning-and-Reinforcement-Learning, UCL/Deepmind, 2018)
- **Mathematics of Deep Learning** (https://joanbruna.github.io/MathsDL-spring18/, NYU Courant Institute, 2018)
- **MIT Deep Learning**( https://deeplearning.mit.edu/, MIT courses and lectures on deep learning, deep reinforcement learning, autonomous vehicles, and artificial intelligence )
- **CMSC 35246 Deep Learning** (https://ttic.uchicago.edu/~shubhendu/Pages/CMSC35246.html, U Chicago Computer Science, 2017)
- **Neural Networks for Machine Learning** by Jeof. Hinton (https://www.cs.toronto.edu/~hinton/coursera_lectures.html, U Toronto, 2012)