10-414/714 – Deep Learning Systems: Algorithms and Implementation

Introduction and Logistics

Tianqi Chen and Tim Dettmers
Carnegie Mellon University

Outline

Why study deep learning systems?

Course info and logistics

Outline

Why study deep learning systems?

Course info and logistics

Aim of this course

This course will provide you will an introduction to the functioning of modern deep learning systems

You will learn about the underlying concepts of modern deep learning systems like automatic differentiation, neural network architectures, optimization, and efficient operations on systems like GPUs

To solidify your understanding, along the way (in your homeworks), you will build (from scratch) needle, a deep learning library loosely similar to PyTorch, and implement many common architectures in the library

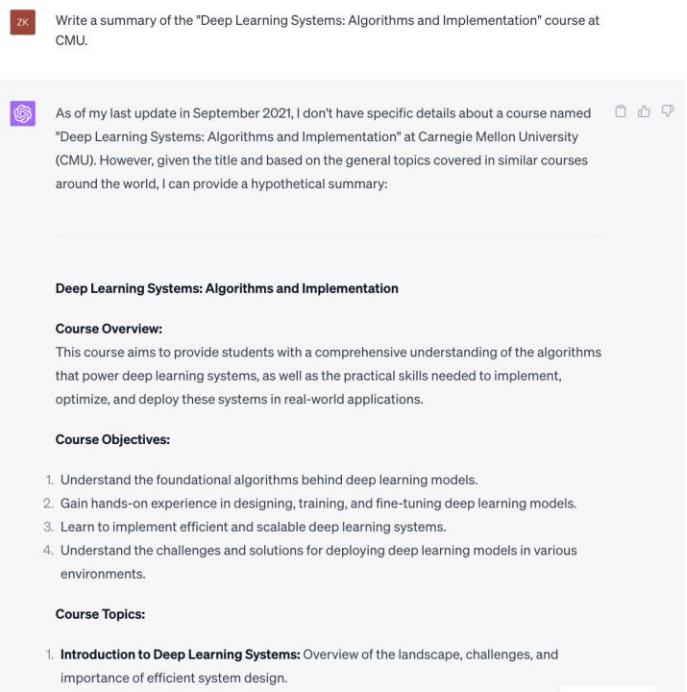
Why study deep learning?

AlexNet (Krizhevsky et al., 2012)

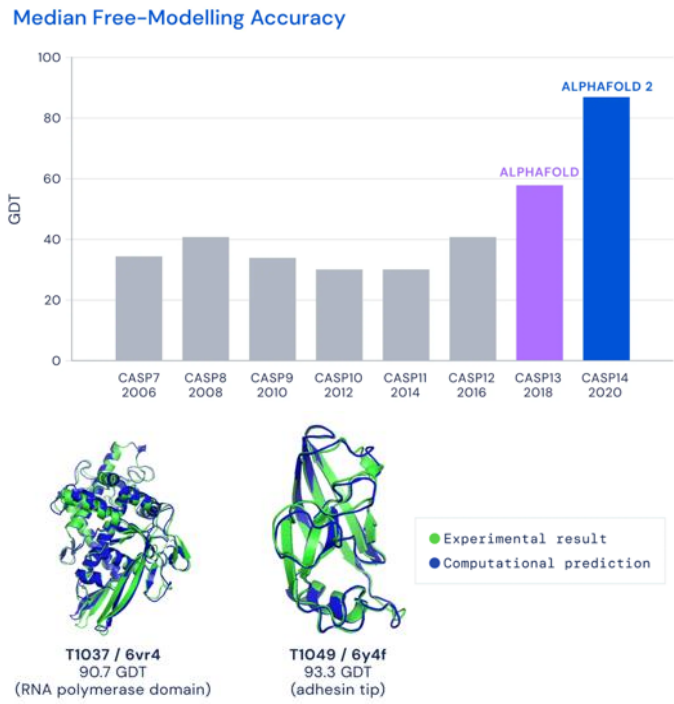
AlphaGo (Silver et al., 2016)

StyleGAN (Karras et al., 2018)

Why study deep learning?



ChatGPT
(OpenAI et al., 2022)



AlphaFold 2 (Jumper et al., 2021)

A dog dressed as a university professor nervously preparing his first lecture of the semester, 10 minutes before the start of class. Oil painting on canvas.

Stable Diffusion
(Rombach et al., 2022)

…Not (just) for the “big players”

https://github.com/ggerganov/llama.cpp

Llama.cpp
(Gerganov, 2023)

https://github.com/huggingface/pytorch-image-models

PyTorch Image Models
(Wightman, 2021)

https://github.com/bitsandbytes-foundation/bitsandbytes

bitsandbytes
(Dettmers, 2021)

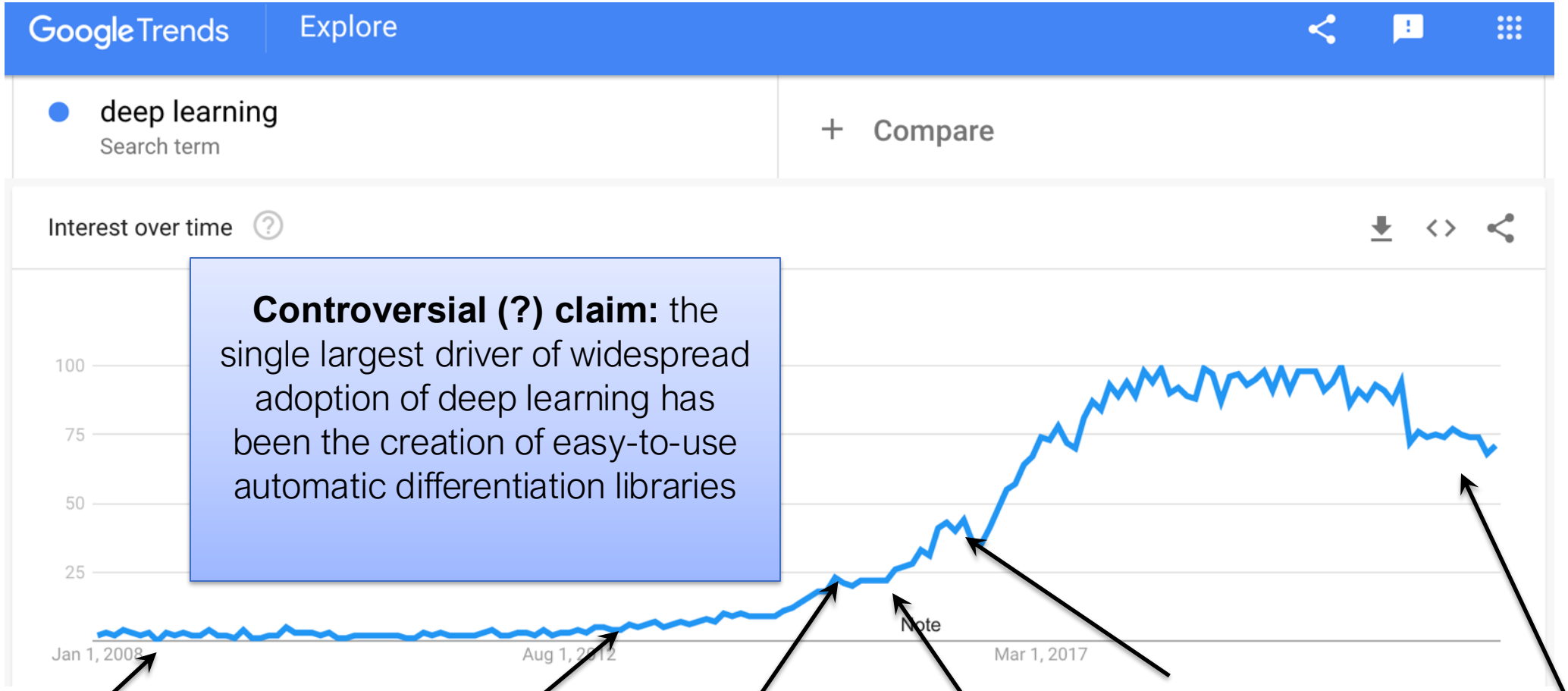
…Not (just) for the “big players”

..many community-driven libraries/frameworks

https://github.com/mlc-ai/xgrammar

XGrammar
(Dong et al., 2025)

Why study deep learning systems?



Reason #1: To build deep learning systems

Despite the dominance of deep learning libraries and TensorFlow and PyTorch, the playing field in this space is remarkably fluid (see e.g., emergence of JAX)

You may want to work on developing existing frameworks (virtually all of which are open source), or developing your own new frameworks for specific tasks

This class (and some practice) will prepare you to do this

Reason #2: To use existing systems more effectively

Understanding how the internals of existing deep learning systems work let you use them *much* more efficiently

Want to make your custom non-standard layer run (much) faster in TensorFlow/PyTorch? … you're going to want to understand how these operations are executed

Understanding deep learning systems is a "superpower" that will let you accomplish your research aims much more efficiently

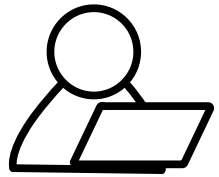
Reason #3: Deep learning systems are fun!

Despite their seeming complexity, the core underlying algorithms behind deep learning systems (automatic differentiation + gradient-based optimization) are extremely simple

Unlike (say) operating systems, you could probably write a “reasonable” deep learning library in <2000 lines of (dense) code

The first time you build your automatic differentiation library, and realize you can take gradient of a gradient without actually knowing how you would even go about deriving that mathematically…

Working on deep learning ten years ago



Researcher

ResNet

Transformer

…

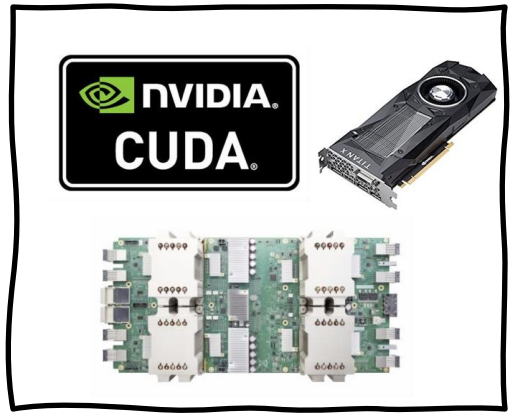
ML Models

44k lines of code

Six months

IMAGENET

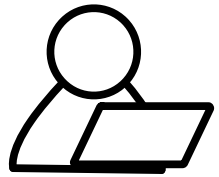
Data



Compute

Based on real story

Working on deep learning now



Researcher

ResNet

Transformer

…

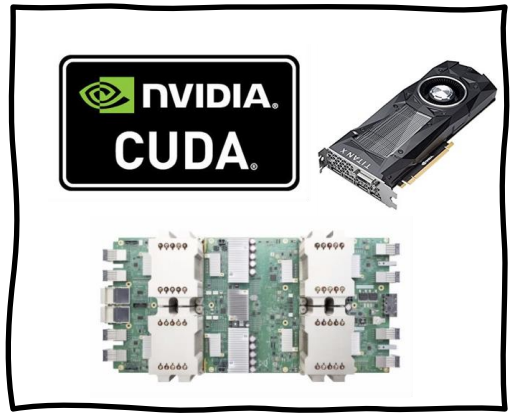
ML Models

100 lines of code

A few hours

Deep learning systems

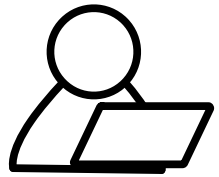
Data



Compute

Based on real story

Working on deep learning with agents



Researcher

ResNet

Transformer

…

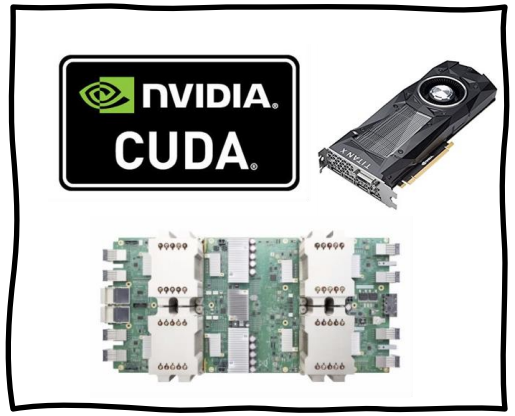
ML Models

500 lines of code

~~A few hours~~ Minutes

Deep learning systems

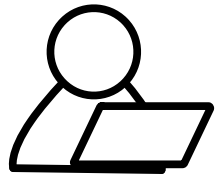
Data



Compute

Based on real story

Working on deep learning (continuously evolving)



Researcher

FLUX deepseek GPT-5

Bigger models

Deep learning systems ?

LAION wikitext

IMAGENET internet

Large high-quality data

More diverse Compute

Elements of deep learning systems

Compose multiple tensor operations to build modern machine learning models

Transform a sequence of operations (automatic differentiation)

Accelerate computation via specialized hardware

Extend more hardware backends, more operators

We will touch on these elements throughout the semester

Outline

Why study deep learning systems?

Course info and logistics

Course instructors

Tianqi Chen

https://tqchen.com/

Professor

catalyst

Carnegie Mellon University
School of Computer Science

Industry past and current

OctoML

NVIDIA

Creator of Major Learning Systems

tvm

mxnet

dmlc XGBoost

Cook and Foodie

Course instructors

Tim Dettmers

catalyst

ML Machine Learning Department

Carnegie Mellon University
Computer Science Department

Ai2

bitsandbytes

QLoRA
4 bit and 8 bit
quantization

Learning objects of the course

By the end of this course, you will …

… understand the basic functioning of modern deep learning libraries, including concepts like automatic differentiation, gradient-based optimization

… be able to implement several standard deep learning architectures (MLPs, ConvNets, RNNs, Transformers), *truly* from scratch

… understand how hardware acceleration (e.g., on GPUs) works under the hood for modern deep learning architectures, and be able to develop your own highly efficient code

Tentative schedule of topics

Date (CMU)	Lecture	Instructor	Slides	Video (2022 version)
8/26	1 - Introduction / Logistics	Dettmers	pdf	YouTube
8/28	2 - ML Refresher / Softmax Regression	Dettmers		YouTube
9/2	3 - Manual Neural Networks / Backprop	Dettmers		YouTube (pt 1) YouTube (pt 2)
9/4	4 - Automatic Differentiation	Chen		YouTube
9/9	5 - Automatic Differentiation Implementation	Chen		YouTube
9/11	6 - Optimization	Dettmers		YouTube
9/16	7 - Neural Network Library Abstractions	Chen		YouTube
9/18	8 - Normalization, Dropout, + Implementation	Dettmers		
9/23	9 - NN Library Implementation	Chen		YouTube
9/25	10 - Convolutional Networks	Dettmers		YouTube
9/30	11 - Hardware Acceleration for Linear Algebra	Chen		YouTube
10/2	12 - Hardware Acceleration + GPUs	Chen		YouTube
10/7	13 - Hardware Acceleration Implementation	Chen		YouTube
10/9	14 - Convoluations Network Implementation	Dettmers		YouTube
10/14	*No class - Fall Break*			
10/16	*No class - Fall Break*			
10/21	15 - Sequence Modeling + RNNs	Dettmers		YouTube
10/23	16 - Sequence Modeling Implementation	Dettmers		YouTube
10/28	17 - Transformers and Autoregressive Models	Dettmers		Youtube
10/30	18 - Transformers Implementation	Dettmers		Youtube
11/4	*No class - Democracy Day*			
11/6	19 - Training Large Models	Chen		YouTube
11/11	20 - Generative Models	Chen		YouTube
11/13	21 - Generative Models Implementation	Chen		YouTube

Listing of lecturers from course website:
https://dlsyscourse.org

Broad topics: ML refresher/background, automatic differentiation, fully connected networks, optimization, NN libraries, convnets, hardware and GPU acceleration, sequence models, training large models, transformers + attention, generative models

(As suggested by course title) lectures are frequently broken down between “algorithm” lectures and “implementation” lecturers (or combined into one)

Prerequisites

In order to take this course, you need to be proficient with:

- Systems programming (e.g., 15-213)
- Linear algebra (e.g., 21-240 or 21-241)
- Other mathematical background: e.g., calculus, probability, basic proofs
- Python and C++ development
- Basic prior experience with ML

If you are unsure about your background, you can talk with the instructors and/or take a look at Homework 0 (released later today); you *should* be familiar with all the ideas in this homework in order to take the course

Components of the course

This course will consist of four main elements

1. Class lectures
2. Programming-based (individual) homeworks
3. (Group) final project
4. Interaction/discussion in course forum

Important to take part in all of these in order to get the full value from the course

Grading breakdown: 55% homework, 35% project, 10% class participation

Class lectures

Class lectures: 11:00-12:20, TR, DH 2315

Lectures will consist of a mix of slide presentations, mathematical notes / derivations, and live coding illustration

Lectures will be recorded but not streamed. Video recordings for (most) lectures available from the previous offering of the course (youtube), and these continue to be available

Slides for lectures will be posted to course web page prior to lecture

In-person Rotation Schedule

We will do in person rotation (IPR)

Half of the class will attend in person, closely watch the announcement on the course forum

We welcome students to come in person as long you prioritize giving seats to students who are assigned to that slot

Hopefully as semester starts we will be able to get seats for most who would like to attend in person

Programming homework assignments

The course will consist of four programming-based homework assignments, plus an additional Homework 0 meant as a review / test of your background

Homeworks are done *individually*, see policies in a subsequent slide

Homeworks are *entirely* coding-based: throughout the assignments you will incrementally develop Needle, a PyTorch-like deep learning library, with: automatic differentiation; gradient-based optimization of models; support for standard operators like convolutions, recurrent structure, self-attention; and (manually-written) efficient linear algebra on both CPU and GPU devices

Homeworks will be autograded using a custom system we are developing for this course (demo and illustration during the next lecture)

Final project

In addition to homeworks, there will also be a final project, done in groups of 2-3 students (exclusively … not in groups of one or four)

Final project should involve developing a substantial new piece of functionality in Needle, or implement some new architecture in the framework (note that you *must* implement it in Needle, you cannot, e.g., use PyTorch or TensorFlow for the final project)

Prior to the final project proposal/team formation deadline, we will post a collection of possible topics/ideas for the project

Class forum

The class will host a forum / chat space on Ed (You should have received the invite, if not, send email to Brynn Edmunds)

Your class participation grade is rated based upon this forum: in order to receive a full credit, you will need to be involved in at least *five* discussions (including, e.g. discussions on homework) on Ed during the course

Top 5 participants in course discussion will also receive additional extra credit for class participation

Collaboration policy

All submitted content (code and prose for homeworks and final project) should be your own content (or written by the group members, for projects)

However, you *may* (in fact are encouraged to) discuss the homework with others in the class and on the discussion forums

- This creates some room for undue copying, but please obey the reasonable person principle: discuss as you see fit, but don't simply share answers

Generative AI Policy

You may use code from generative AI tools (e.g., Cursor, Claude Code, Codex CLI, etcetera), no need to cite or specify it was from these tools

You are ultimately responsible for anything the tools generate, including any flaws this code may contain

You are responsible for how much you learn from assignments when you solve them with AI tools

I would strongly recommend completing HW0 without the tools: it's meant to be a warmup assignment (honestly, these tools will be able to complete it easily), but the course will be very challenging later, particularly for the project, if you can't complete these yourself

Tips on how to use and learn with AI tools

Multi-phase procedure to work with coding agents (Claude Code, Codex CLI):

1. Gather relevant data from web, other solutions online; if you do not understand parts of the assignment, ask ChatGPT to explain it
2. Insert knowledge into the environment (.md files + prompt + images)
3. Solve assignment. Have the agent write software tests that test correctness (beyond existing tests)
4. Interact with coding agent to understand details of implementation. Then write session details to file/environment.
5. Follow procedure 1-4, but this time have the agents design a quiz for you to test your understanding of the solution + discussion from coding agent.
6. Try to get 90% correct, then carefully study the code to see if there are any gaps in your understanding
7. Generate more questions to test your understanding

Signs of poor learning with AI tools

AI tools only slightly better than your understanding. If your understanding is limited, then the project will be difficult

Mastery needed to be able to work in this field. AI tool use can prevent mastery

If you are not able to answer "why" questions, you probably should slow down AI tool use and do an in-depth session to increase understanding

If you cannot answer details, your understanding is too shallow (unable to debug problem; unable to describe what a solution should look like).

Student well-being

CMU and courses like this one are stressful environments

In our experience, most academic integrity violations are the product of these environments and decisions made out of desperation

Please don't let it get to this point (or potentially much worse); contract the instructors/Tas ahead of time if you feel that issues are coming up that are interfering with your ability to participate fully in the course

Don't sacrifice quality of life for this course: make time to sleep, eat well, exercise, be with friends/family, socialize, etc

In the remaining time…

Log on to Ed https://edstem.org/us/courses/81478/discussion and say hello

Checkout the course homepage https://dlsyscourse.org/