Introduction to Deep Learning

0. Logistics

Spring 2022

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

Who is this course designed for?

- Students from any background that want to learn deep learning
- Students who are willing to put in 20 hours a week on this course
- Students who give continuous feedback and engage on Piazza
- Students who are mature and want to be challenged
- Students who want to be ready for AI research & engineering roles

Examples of Successful People

Architects

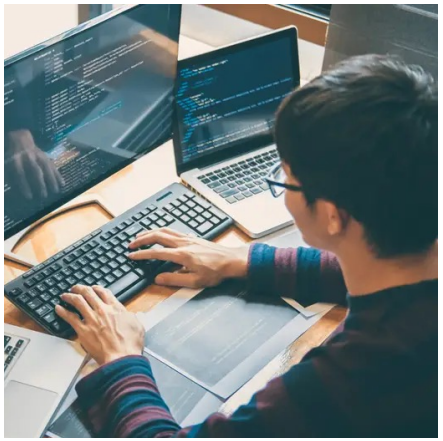
Artists

Social Scientists

Hostage Negotiators

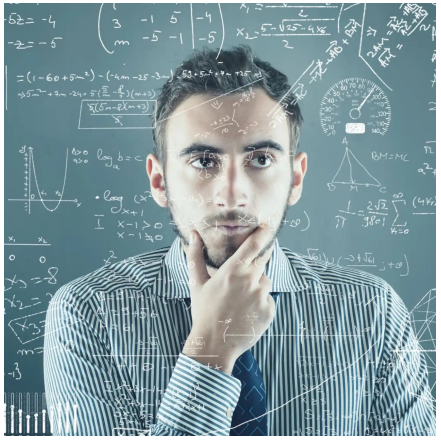
Examples of Successful People

Software Engineers

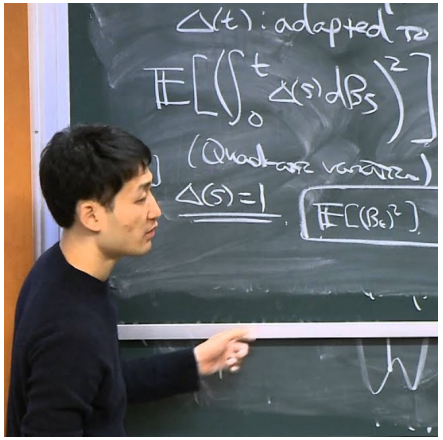


Computer Scientists

Data Scientists



Quantitative Analysts



Research Specialties: Computer Vision, Natural Language, Speech, and Audio

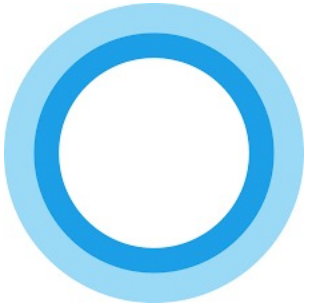
Students From MANY Departments

- ECE
- LTI
- CS
- ISM
- MEG
- SCS
- BUS
- STA
- ICT
- MSC
- BSC
- MUS
- PHY
- PPP
- ROB
- INI
- BCA
- HCP
- CMY
- AIT
- ITM
- CEE
- HSS
- BMD
- IPM
- GBS
- IPS
- and more!

Neural Network Examples

- Siri, Alexa, Google Home, Cortana
  - Unsupervised Speech Recognition (USR)
  - Question Answering (BERT)



- USR: arxiv.org/abs/2105.11084
- BERT: arxiv.org/pdf/1810.04805.pdf

Neural Network Examples

- NVIDIA DLSS and AMD FSR

Real-ESRGAN: github.com/xinntao/Real-ESRGAN
NVIDIA DLSS: youtu.be/SBjL0M25t04
AMD FSR: youtu.be/eHPmkJzwOFc

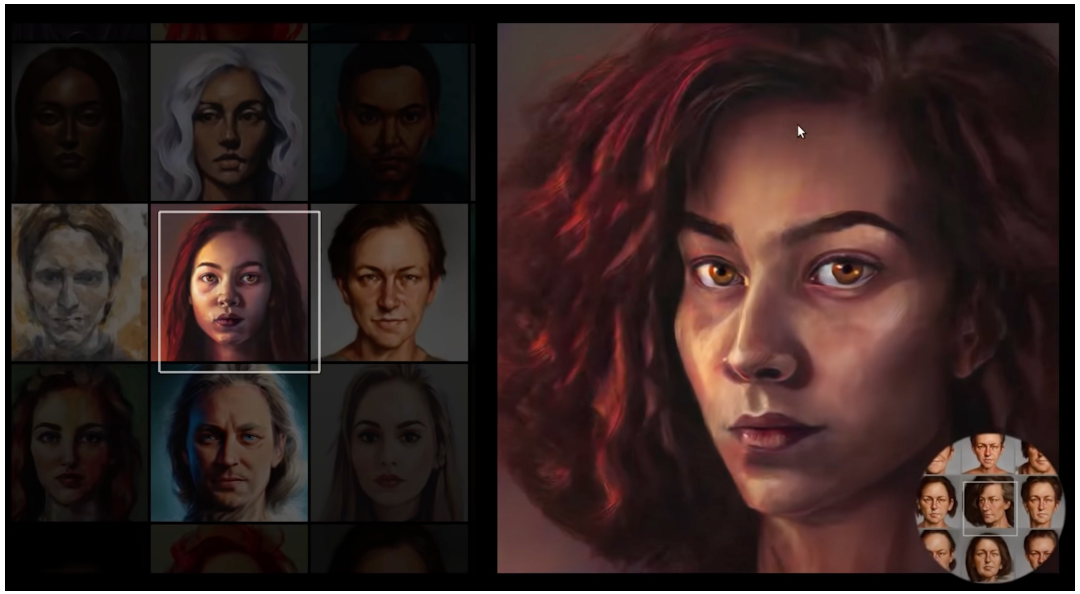
Neural Network Examples

- Object Detection: Detectron

github.com/facebookresearch/detectron2

Neural Network Examples

- Other Creative Work: StyleGan2



youtu.be/9QuDh3W3lOY | github.com/NVlabs/stylegan2

Preparation For The Course

- Required Classes:
  - Fundamentals of Programming (or equivalent)
- Highly Recommended Classes:
  - Calculus and Linear Algebra
- Nice To Have:
  - Vector Calculus and Software Engineering
- We will teach the math you need to know when it comes up, but it takes time to learn.

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

Who should not take this course?

- Not for students who want an easy A
- Not for students who don’t want to work with others
- Not for students who do not ask for help
- Not for students who put off deadlines to last minute
- Not for students who want to be given all the answers

Unacceptable Behavior

- Submitting code copied from GitHub or students
- Not engaging or contributing with project groups
- Disrespecting other students or TAs
- Complaining or making fun of Bhiksha's disability

Job Markets & Neural Networks

This guy didn't know about neural networks (a.k.a deep learning)

This guy learned about neural networks (a.k.a deep learning)

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

Learning Outcomes

- We teach you the **engineering** behind *creating deep learning frameworks* from scratch
- We teach you the **science** behind *training deep learning models* from scratch
- We prepare you for deep learning **interviews** and *build projects for your resume*
- We teach you the **pre-requisite knowledge** needed for *deep learning research*
- And we teach you practical aspects of what **compute resources** cost, how to use them *efficiently*, and the options you have available for different problems.

Topics and Content Overview

- Conceptual Knowledge
  - Some historical perspective
  - Types of neural networks and underlying ideas
  - Learning in neural networks
    - Training, concepts, practical issues
  - Architectures and applications
  - Will try to maintain balance between squiggles and concepts (concept >> squiggle)

- Practical Knowledge
  - Familiarity with training
  - Implement various neural network architectures
  - Implement state-of-art solutions for some problems

- Overall: Set you up for future deep learning research in your area

Topics and Content Overview

- Basic network formalisms:
  - Multilayer Perceptrons
  - Convolutional Neural Networks
  - Recurrent Neural Networks (RNNs)
  - Boltzmann Machines
- Some advanced formalisms
  - Generative models: VAEs
  - Adversarial models: GANs
  - Graph Neural Networks
- Topics we will touch upon:
  - Computer Vision: Recognizing Images
  - Text Processing: Modelling and Generating Language
  - Machine Translation: Sequence-to-Sequence Modelling
  - Modelling Distributions and Generating Data
  - Speech Recognition
  - Wish List: Reinforcement Learning and Games

Course Philosophy

- Our goal is to **teach** you deep learning. You are **graded** on your ability to **show** you understand deep learning.
- We give you many many opportunities to **demonstrate** your understanding. You will have multiple opportunities to **work** towards an A and **improve** on your shortcomings IF you are willing to put in the work.
- We tell you when you are wrong, why we think you are wrong, and **how you should think** about getting the right answer.
- We don't just want you to have deep learning knowledge, we want you to be able to **think critically and creatively** about designing solutions to new problems.

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork and Study Groups

Instructors and TAs

- Instructors:
  - Bhiksha Raj <bhiksha@cs.cmu.edu>
  - Rita Singh <rsingh@cs.cmu.edu>
- Teaching Assistants:
  - Contact Info On Course Page
  - We have TAs from:
    - Pittsburgh Campus
    - Silicon Valley Campus
    - Kigali Campus
    - Doha Campus
  - Remote and in-person OH offerings and times are subject to change
- Office Hours Listed On Course Page:

deeplearning.cs.cmu.edu

Logistics: Lectures..

- *In-class lectures (unless rules change due to covid)*
  - Lectures are also recorded
  - Recordings will be posted

- Important that you view the lectures
  - Even if you think you know the topic
  - Your marks depend on viewing lectures
  - We will monitor attendance (more on this later)

Additional Logistics

- Reading and Engaging on Piazza is Mandatory
  You will fail quickly if you do not use Piazza.

- Compute Infrastructure:
  - Everyone gets 3 Amazon Web Service coupons
  - Each coupon is worth $50, for a total of $150
  - You must maintain progress to receive a coupon, and we distribute them after we reveive them.

Lecture Attendance

- You get marks for attendance
  - Our performance metrics over the semesters show a distinct correlation between attendance and course scores
  - We also note a distinct *inverse* correlation between attendance and the amount of help you require on piazza and during office hours
  - To encourage attendance, we assign 1 mark for attendance
    - 1% of your total grade for 11685/11785/18786
    - 1.3333% of your total grade for 11485
    - This can be the difference between a B and an A

- We will track lecture attendance
  - More on next slide

Lecture Schedule

- On website
  - The schedule for the latter half of the semester may vary a bit
    - Guest lecturer schedules are fuzzy..
- Guest lectures:
  - TBD
    - One or more of: Scott Fahlman, Shinji Watanabe, Gerald Friedland, Graham Neubig

Recitations

- We will have 14 recitations
- Will cover implementation details and basic exercises
  - Very important if you wish to get the maximum out of the course
- Topic list on the course schedule

- *Strongly recommend attending all recitations*
  - *Even if you think you know everything*

Recitations Schedule

- Every Friday of the semester
- See course page for exact details!

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

Evaluation

- Homework Part 1 – Building DL Framework
- Homework P2 – Building Competitive Model
- Quizzes – Testing DL Understanding
- Project – Demonstrate DL Research Ability

- See course website for syllabus and weights.

deeplearning.cs.cmu.edu

Weekly Quizzes

- 10 multiple-choice questions
- Related to topics covered that week
  - On both slides and in lecture
- Released Friday, closed Sunday night
  - This may occasionally shift, don't panic!
- There will be 14 total quizzes
  - We will consider the best 12
  - This is expected to account for any circumstance-based inability to work on quizzes
    - You could skip up to 2

Lectures and Quizzes

- Slides often contain a lot more information than is presented in class
- *Quizzes will contain questions from topics that are on the slides, but not presented in class*
- *Will also include topics covered in class, but not on online slides!*
- There will be questions based on latest research papers in the quiz – the links to the papers will be provided

Homeworks

- There will be one early homework (released before the start of the semester) and four in-term homeworks
  - Homework 0: Preparatory material for the course
  - Homeworks 1-4: Actual neural-net exercises

- Homeworks 1-4 all have two parts:
  - Part 1: Autograded problems with deterministic solutions
    - You must upload them to autolab
  - Part 2: Open problems posted on Kaggle

- Bonus HWs:
  - There will be 3 bonus homeworks corresponding to HW1-3 and 4 separate autograd bonus homeworks
  - These marks will not contribute to final grading curves but give you the chance to make up for marks missed elsewhere

Homeworks 1-4 – Part 1

- Part 1 of the homeworks evaluate your ability to code in neural nets on your own from scratch
  - If you implement all mandatory and bonus questions of part 1 of all homeworks, you will, hopefully, have all components necessary to construct a little neural network toolkit of your own
    - "mytorch" ☺️
- The homeworks are autograded
  - Be careful about following instructions carefully
    - The autograder is setup on a computer with specific versions of various packages
    - Your code must conform to their restrictions
  - If not the autograder will often fail and give you errors or 0 marks, even if your code is functional on your own computer

*slack days (mentioned later) do not apply to part 1s

Homeworks 1-4, Part 2

- Part 2 of every homework tests your ability to solve complex problems on real-world data sets
  - The early submission deadline (ESD) is required and worth 10/100 points
  - The remaining 90/100 points are determined by your final score relative to the cutoffs
  - Cutoffs are determined after ESD, and your score is linearly interpolated between cutoffs
  - There are 4 standard cutoffs giving 90 pts, 70 pts, 50 pts, 30 pts, and 0 pts
  - There is a 5th bonus cutoff for extra credit determined by the highest scoring student.

Homework Deadlines

- Multiple deadlines
- Separate deadline for Autograded deterministic component
- Kaggle component has multiple deadlines
  - *Early Submission D*eadline : Worth 10% of P2 and affirms you have started the assignment
  - *On-time Submission Deadline:* Your final submission must occur before this deadline to be eligible for full marks
  - *Late / Slack Deadline:* Submissions after the On-time Submission Deadline will receive a penalty, and you can use slack days to avoid a penalty (but not to get bonus)
- Homeworks: Late policy
  - Everyone gets up to 7 total slack days (does not apply to initial submission or part 1 of the HWs)
  - You can distribute them as you want across your HWs
    - You become ineligible for "A+" bonus if you're using your grace days for Kaggle
  - Once you use up your slack days, all subsequent late submissions will accrue a 10% penalty (on top of any other penalties)
  - There will be no more submissions after the drop-dead deadline
  - Kaggle: Kaggle leaderboards stop showing updates on full-submission deadline
    - But will continue to privately accept submissions until drop-dead deadline

- ***Please see course webpage for complete set of policies***

Course project

- If you're taking 11-785/18-786, you will be required to do a course project
  - 11-685 students will be assigned a fifth HW that is equivalent to a project
- Projects are done by teams of students
  - Ideal team size is 4
    - 11-685 teams are ideally 2-person teams
  - You are encouraged to form your teams early
- Projects are intended to exercise your ability to comprehend and implement ideas beyond those covered by the HWs
- Project can range from
  - Implementing and evaluating cutting-edge ideas from recent papers
    - *Verifying* results from "hot" published work
  - "Researchy" problems that might lead to publication if completed well
  - Proposing new models/learning algorithms/techniques, with proper evaluation
  - Etc.

Course project

- Think about forming project teams as soon as possible. More details on website.
  - If you don't form your own teams, we will team you up
- Each team must:
  - Submit a project proposal by the deadline listed on the website
  - Submit a mid-way report ¾ way through the semester
  - Submit a preliminary full report three days **before the presentation due date**
  - Make a **5 min** video presentation of the project at the end of the semester
    - Can be presented by one, some, or all team members
    - Will be evaluated by the instructor, TAs, and your classmates
    - Ensure you explain the problem, proposed solution, and the evaluation clearly
    - Allocate enough time to make the presentation, it is not as easy as you think
    - Poor presentation can significantly affect your project score :)
  - Submit a final full report at the end of the semester
  - Defend your project in front of peers and TAs
  - Templates for proposals and reports will be posted
- *Each team will be assigned a mentor from among the TAs, who will monitor your progress and assist you if possible.*
- **More details on project evaluations will be posted towards the end of the sem**
- **The project is often the most fun portion of the course**

Grading

Weekly Quizzes		**24**
14 Quizzes, bottom two dropped		24
Assignments		**50**
HW1 – Basic MLPs	(AL + Kaggle)	12.5
HW2 – CNNs	(AL + Kaggle)	12.5
HW3 – RNNs	(AL + Kaggle)	12.5
HW4 – Sequence to Sequence Modelling	(AL + Kaggle)	12.5
Team Project (Not for 11-485)		**25**
Proposal		-
Mid-term Report		5
Preliminary Full Report		-
Project Presentation		10
Peer Reviewing		binary multiplier
Final report		10

Don’t forget: There is also 1 mark for attendance

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

Out Roles As TAs: HW Part 1

- Architecture Diagram w/ Example Data
- Solutions Created And Verified Against PyTorch
- Write Mathematics Consistent With Solution
- Create Write-up w/ Diagram, Math, and Redacted Code
- Verify Other TAs Get Same Solutions Given Math
- Create AutoLab w/ Type, Shape, and Value Verification
- Finalize Assignment and Answer Student Questions

Your Roles As Students: HW Part 1

- Read the write-up to understand what is expected
- Work through the examples in the write-up by hand
- Write the code to reproduce the same examples
- Consider what changes are needed for arbitrary examples
- Meet regularly with your study group to discuss challenges
- Ask questions on Piazza if you get stuck at any stage

Out Roles As TAs: HW Part 2

- Obtain, Prepare, Partition, and Upload Data With Labels
- Create Standard Dataset Class Diagram w/ Example Data
- Identify, Train, and Validate “Good” Model Architectures (private)
- Create Efficient Architecture And Train With Verbose Logs (public)
- Verify Other TAs Get Same “Good” Solutions Given Logs (private)
- Verify Other TAs Get Same Efficient Solutions Given Logs (public)
- Finalize Assignment and Answer Student Questions

Your Roles As Students: HW Part 2

- Read the write-up to understand what is expected
- Watch and work through recitations starter code
- Identify and implement models in literature that perform well
- Perform an ablation study of different training specifications
- Meet regularly with your study group to discuss challenges
- Ask questions on Piazza if you get stuck at any stage

Getting the most of Piazza

- You should engage on Piazza regularly. If you are stuck on a problem for more than 20 minutes, then you should post on Piazza to get feedback to help you get unstuck.
- If you don't get the response you wanted on Piazza, 9 times out of 10 this is because:
    1. You did not explain what you tried and the problem is missing context.
    2. It is unclear how what you observed is different than what you expected.
    3. You did not provide enough information to reproduce the problem.
    4. You did not watch lecture, did not watch recitation, did not read the handout, or did not read instructor posts relevant to your problem understanding and setup.

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

Course Deliverables

- Course is implementation heavy
  - A lot of coding and experimenting
  - Will work with some large datasets

- Language of choice: Python
- Toolkit of choice: Pytorch
  - You are welcome to use other languages/toolkits, but the TAs will not be able to help with coding/homework
    - Some support for TensorFlow

- We hope you have gone through
  - Recitation zero
  - HW zero
    - Carries no marks

Teamwork

- Learning happens best together
  - You will learn more from each other than you will from us
- *We encourage teamwork*
  - But there are strict rules…

Study groups

- Please form study groups
- If you do not have a study group of your own, we will form one for you
  - Please register on the forms posted on Piazza
- *Everyone must be part of a study group*

Study groups

- What study groups may do:
  - Discuss homework problems and solutions
  - Discuss papers
  - Discuss class work
  - Discuss quizzes
- We encourage you to meet regularly to discuss IDL work
- Study groups may also go on to form project teams
- So what are the caveats? What may you *not* do

Study groups vs. cheating

- Every student must solve their quizzes by themselves
  - You may discuss the questions with your study groups/friends, but when you solve the quiz, isolate yourself and do it alone
- Every student must solve every homework by themselves
  - You may discuss the homeworks with your friends, and even help them debug their code, but when you finally solve it, every line of your code (except libraries that have been okayed by course staff) must be written by you
  - Your solution must be yours
- Plagiarizing code from the web or your friends constitutes cheating
  - And submitting solutions not obtained by you constitutes cheating

Cheating

- You are here to learn DL yourself, not to demonstrate how well your friend, or that guy on the web has learned DL
- You are at CMU – which means you are among the brightest and best students in the *world*
  - You probably were among the top students in your peer group all your life, before you came here
  - It will be an insult to yourself and everything you ever stood for in your life to lower yourself from your own standards and start cheating
    - So don't!!!
- If you are unsure whether something you're doing constitutes cheating or not, check with us

Mentoring

- Every study group will be assigned a TA mentor
  - We will track your progress and reach out to you if you appear to be in trouble
- If in trouble, reach out to your TA mentor and/or the instructor
  - If you feel you're falling behind, reach out
  - If you feel you are struggling, reach out
  - If you feel pressured/unable to cope, reach out
  - We will try our best to help you
  - *Please watch Recitation 0I if you are stuck or feeling overwhelmed*
- We aim to make this a successful course for all of you
  - In our ideal world, everyone performs well enough to get an A
  - *Without lowering our standards* – i.e. we would like to bring *you* all up to where we believe you deserve an A
  - Everything about this course is geared to that objective

Outline

- Who should take this course?
- Who should not take this course?
- Learning Outcomes and Content Overview
- Course Logistics
- Scores, Cutoffs, and Grades
- Student and TA Expectations
- Teamwork, Study Groups, and Cheating
- Challenges

This course is not easy

- A lot of work!

This course is not easy

- A lot of work!
- *A lot of work!!*

This course is not easy

- A lot of work!
- *A lot of work!!*
- ***A lot of work!!!***

This course is not easy

- A lot of work!
- *A lot of work!!*
- ***A lot of work!!!***
- ***A LOT OF WORK!!!!***

Not for chicken!

This course is not easy

- A lot of work!
- *A lot of work!!*
- ***A lot of work!!!***
- ***A LOT OF WORK!!!!***

But somewhat calibrated (over the years) to ensure it is doable

Over 60% of students got some flavor of A each of the past three semesters and they deserved it

This course is not easy

- A lot of work!
- *A lot of work!!*
- ***A lot of work!!!***
- ***A LOT OF WORK!!!!***
- *Mastery-based* evaluation
  - Quizzes to test your understanding of topics covered in the lectures
  - HWs to teach you to implement complex networks
    - And optimize them to high degree
- Target: Anyone who gets an "A" in the course is technically ready for a deep learning job

HW0 / Recitation 0

- Please, please, please, please, please, please go through the videos for recitation 0, and complete HW0.
  - These are essential for you to gain comfort with the coding require in the following homeworks
- HW1 part 1 also has many components intended to help you *later* in the course
  - So if it seems a bit dense, please bear with it, its worth it
- HW1 is the easiest HW!

Questions?

- Please post on piazza