CS 4644 / 7643: Deep Learning

Prof. Danfei Xu

School of Interactive Computing
Georgia Tech

Website

- https://sites.cc.gatech.edu/classes/AY2026/cs7643_fall/
- Short URL: https://tinyurl.com/cs7643-fa25
- Linked on Canvas

Are you at the right place?

- This is CS 4644(DL) / CS 7643
  - “On campus” class

- This is NOT CS 7643-O01/OAN/Q/R
  - Online class for OMSCS program (Prof. Zsolt Kira)

Fall 25 Delivery Format

- In-person
  - Clough UG Learning Commons 144
- Recording
  - We STRONGLY encourage you to attend the lectures in person.
  - Lectures are recorded and available for viewing
- **Remember: Content is free online.**
  - **You are here for the interactive experience.**

Outline for Today

- What is Deep Learning, the field, about?
- What about GPT-5/Genie3/Pi-0… ?
- What is this class about?
    - What to expect?
    - Logistics
- FAQ

Survey

Undergrad?

M.S.?

Ph.D.?

CS (CoC) / ECE?

Other Engineering?

Math / Natural Science?

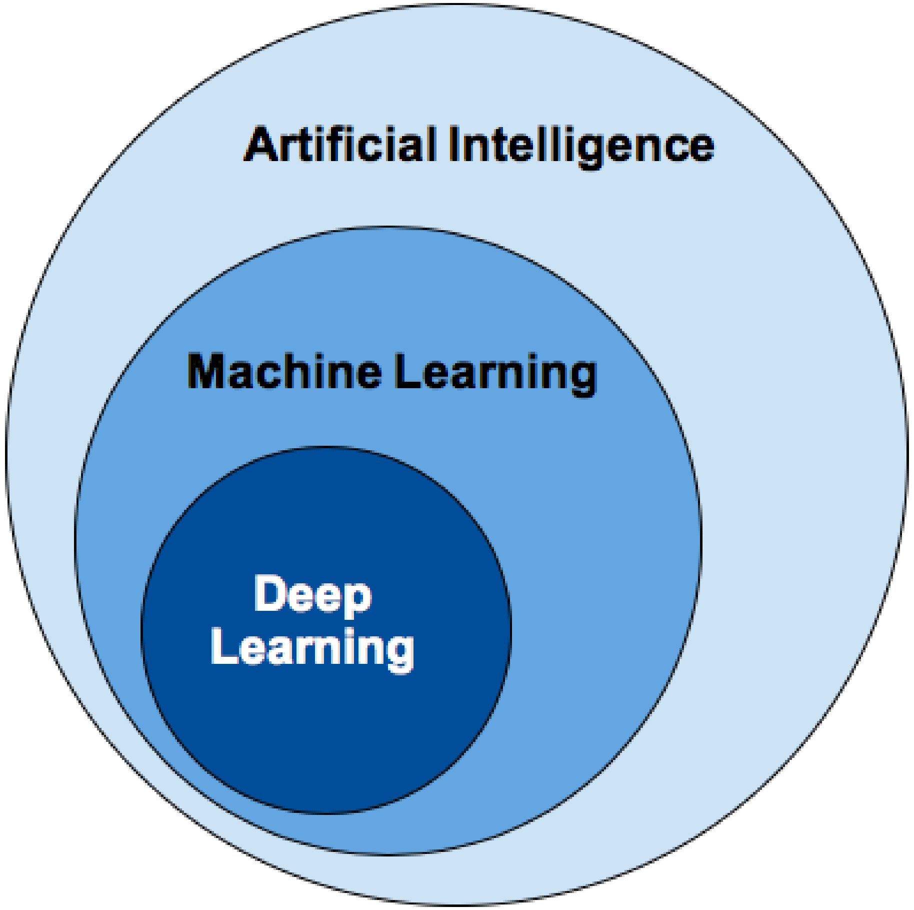
Business?

Others?

Outline

- What is Deep Learning, the field, about?
- What is this class about?
  - What to expect?
  - Logistics
- FAQ

Concepts



"Deep Learning is part of a broader family of **machine learning methods** based on **artificial neural networks**"

--- https://en.wikipedia.org/wiki/Deep_learning

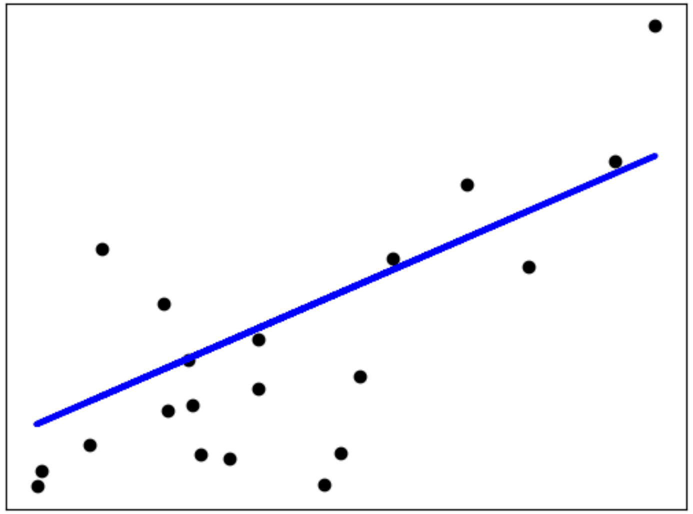
Image Credit: https://www.sumologic.com/blog/machine-learning-deep-learning/

What is Machine Learning?

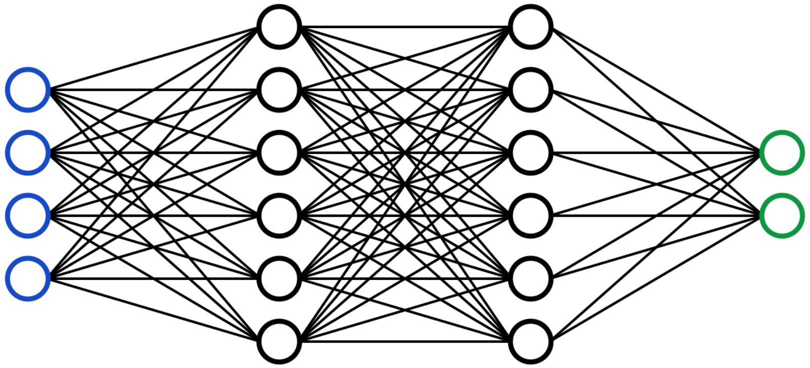
Enable machines to exhibit intelligent behaviors or improve their performances through data, without being explicitly programmed.

For example:

Linear regression is to find an optimal linear function (**model**) for a given dataset such that the difference between the predicted values and the actual values are minimized (**objective**).



So what is Deep Learning?



- **Model**: (Deep) Artificial Neural Networks
- **Objective**: Representation Learning
  - Automatically discover useful features/representations for a **task** from **raw data**
- **Learning Method**: Ways to train the neural networks to optimize the objectives
  - Supervised/Unsupervised/Reinforcement/Generative/… Learning
- **System**: Software (PyTorch/Jax/TensorFlow…) and hardware (GPU, cluster, edge devices…)
- **Simply**: Deep Learning

Studying Deep Learning in 2025

How is Deep Learning related to GPT5/Genie3/MidJourney/Pi0/ … (insert the AI models you saw in the news yesterday)?

- Deep learning is the **core technology** behind today's most advanced AI systems
- Modern breakthroughs like LLMs, image generators, and voice assistants are all built on **deep neural networks**
- The current AI revolution is fundamentally the result of **scaling up** these deep learning techniques with *massive datasets, enormous computational power*, and *architectural & algorithmic innovations*

So what is Deep Learning

Ways to think about Deep Learning:

- Bottom-up: (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

- Top-down: End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extraction

Hierarchical Compositionality

VISION

pixels → edge → texton → motif → part → object

NLP

character → word → NP/VP/.. → clause → sentence → story

Composing Simple Functions to Build Complex Functions

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Building A Complicated Function

Given a library of simple functions

$\sin(x)$

$\log(x)$

$\cos(x)$

x^3

$\exp(x)$

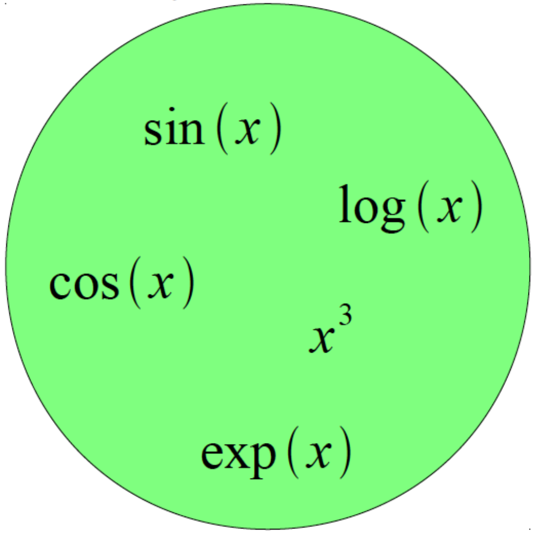
Compose into a

complex function

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Building A Complicated Function

Given a library of simple functions



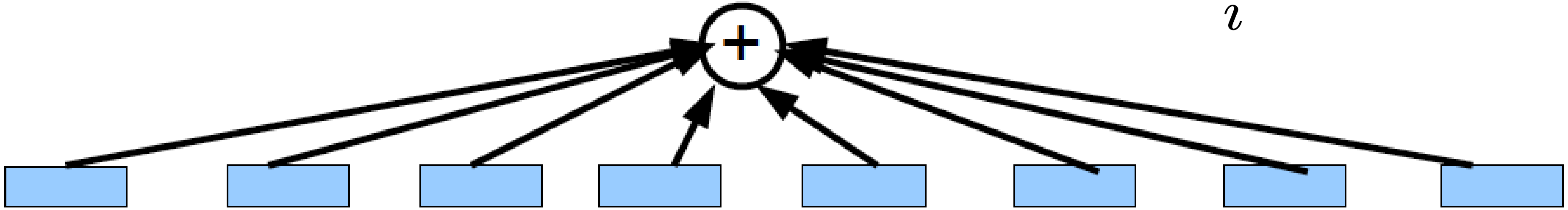
Compose into a

complex function

Idea 1: Linear Combinations

- Boosting
- Kernels
- …

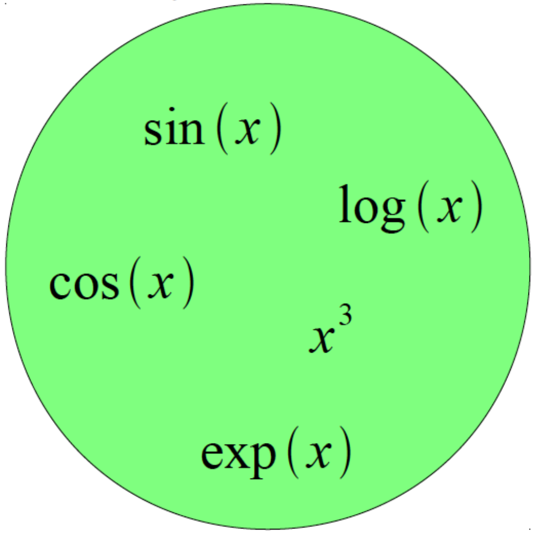
$$f(x) = \sum_i \alpha_i g_i(x)$$



Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Building A Complicated Function

Given a library of simple functions



Compose into a

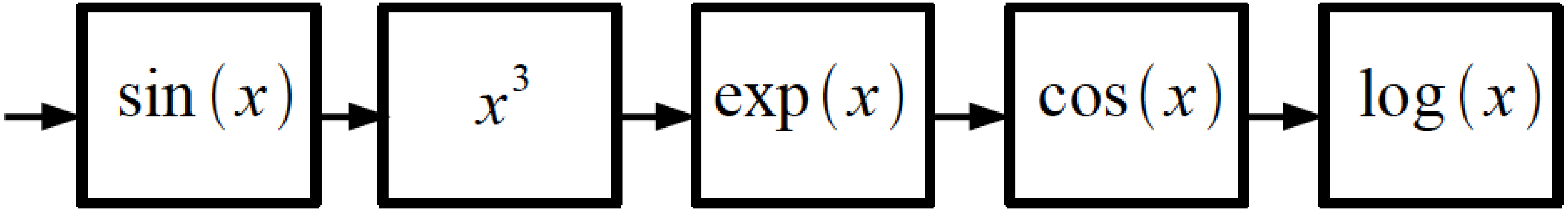
complex function

Idea 2: Function Composition

$h = g \cdot f$ such that $h(x) = g(f(x))$

Can we make it more expressive?

$$f(x) = \log(\cos(\exp(\sin^3(x))))$$



Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Building A Complicated Function

Given a library of simple functions

$\sin(x)$

$\log(x)$

$\cos(x)$

x^3

$\exp(x)$

Compose into a complex function

Idea 3: Layer Composition

Compose a set of layers **(parametric functions)** through which the input data get transformed.

More layers = "Deeper"

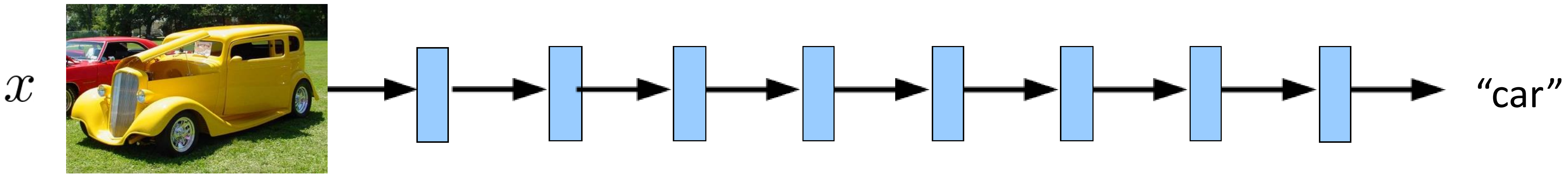
Production models today have **billions** of learnable parameters

Parametric functions

$$f_\theta(x) = \boxed{g_{\theta_n}}(\ldots g_{\theta_2}(g_{\theta_1}(x)\ldots))$$

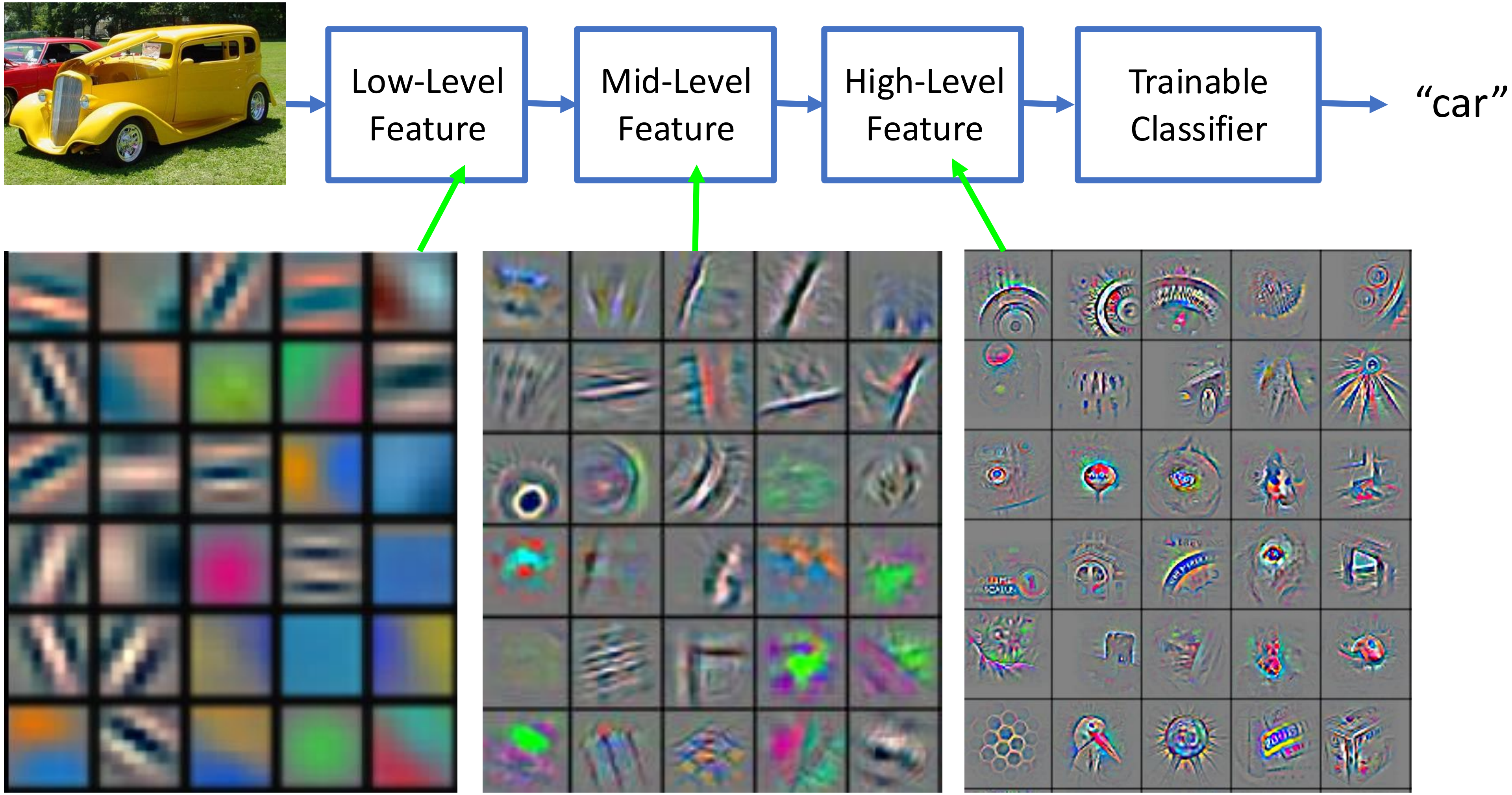
Linear Layer:
$$g(x) = Ax + b$$

Nonlinear Layer:
$$g(x) = \max(0, x)$$



Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

Deep Learning = Hierarchical Compositionality



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

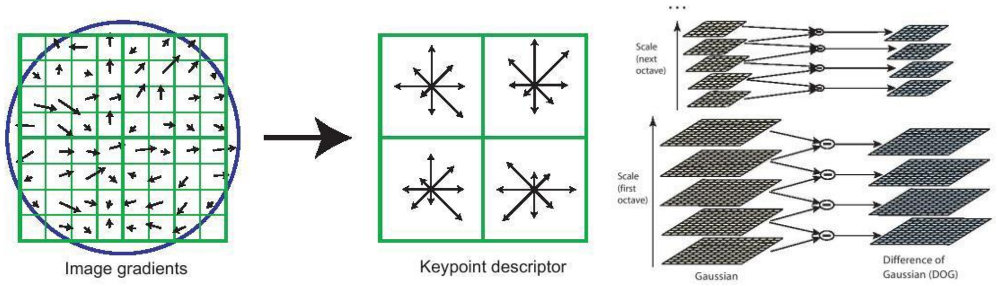
So what is Deep Learning

Ways to think about Deep Learning:

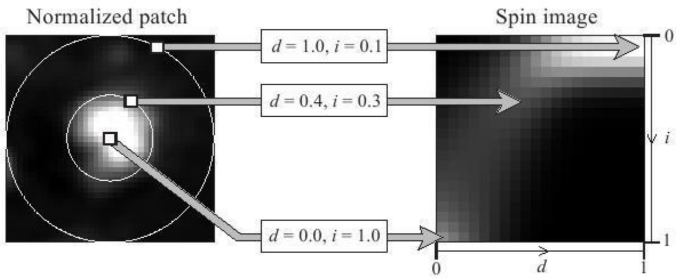
- Bottom-up: (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

- Top-down: End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extraction

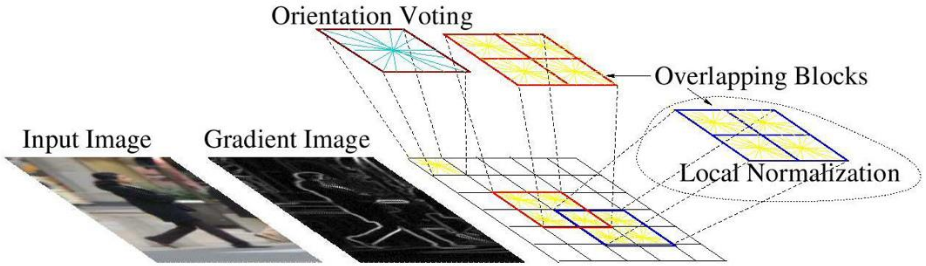
Pre-deep learning: Feature Engineering



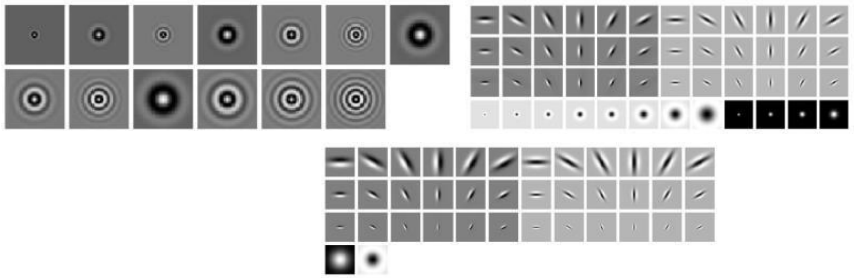
SIFT



Spin Images



HoG



Textons

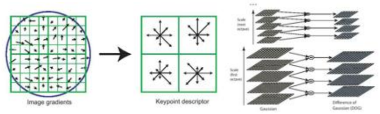
and many many more….

"Shallow" vs Deep Learning

- "Shallow" models

hand-crafted

Feature Extractor



fixed

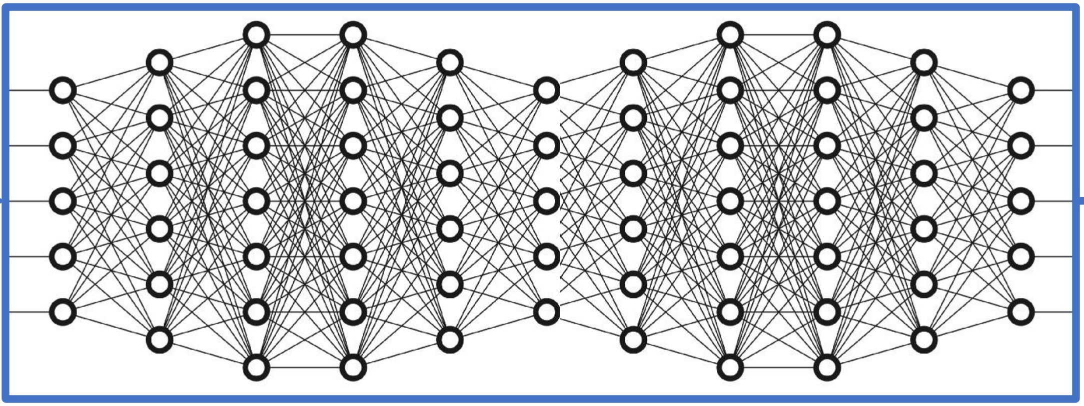
"Simple" Classifier

(linear regressor, SVM, ...)

learned

"Car"

- Deep models



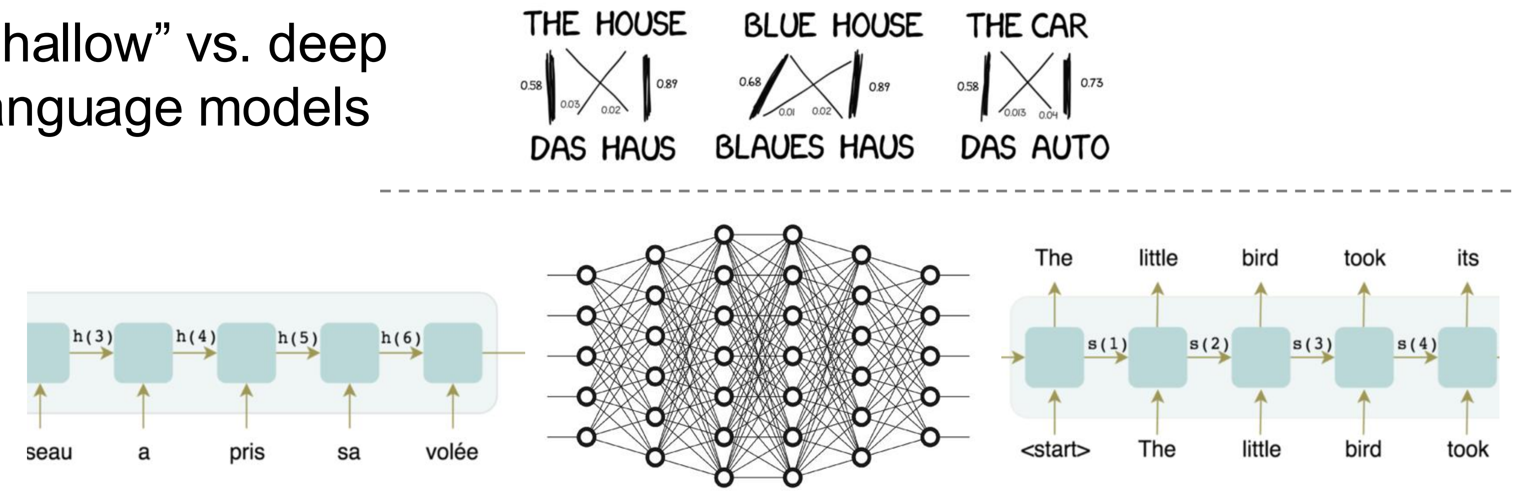
Learned

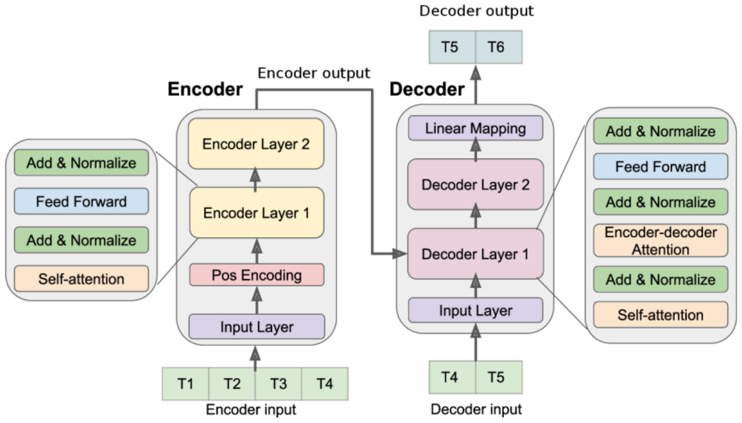
"Car"

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

“Shallow” vs Deep Learning

“Shallow” vs. deep language models



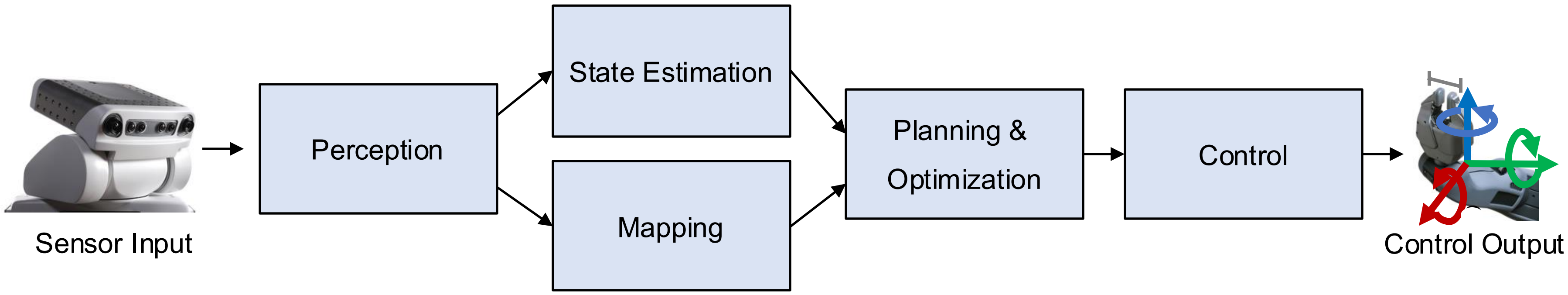


Transformer Models
(Vaswani *et al.*, 2017)

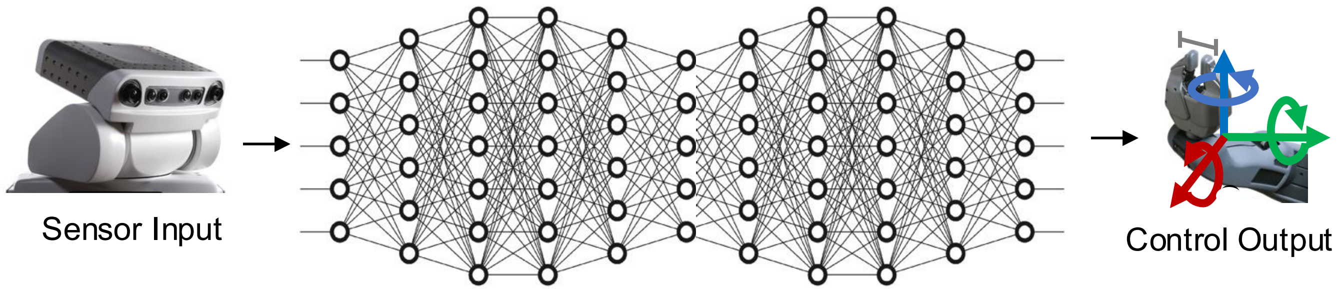
GPT4 large language model
(OpenAI 2023)

https://www.freecodecamp.org/news/a-history-of-machine-translation-from-the-cold-war-to-deep-learning-f1d335ce8b5/

“Pipelining” vs. “End-to-End Learning”



Hand-engineered pipelines



End-to-end learning
(“pixel-to-torque”)

So what is Deep Learning

Ways to think about Deep Learning:

- Bottom-up: (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

- Top-down: End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extraction

Benefit of Deep Learning

- (Usually) Better Performance
  - Caveats: given enough data, similar train-test distributions, non-adversarial evaluation, etc, etc.

- Apply to new domains without expertise knowledge
  - RGBD/Lidar
  - Language data
  - Gene-expression data
  - Complex controlling problem
  - Unclear how to hand-engineer

- New abilities emerge with more data and compute

- “Homogenization” of model design

“Expert” intuitions can be misleading

- *“Every time I fire a linguist, the performance of our speech recognition system goes up”*
  - Fred Jelinik, IBM ’98

- *“Because gradient descent is better than you”*
  - *Yann LeCun,* CVPR ‘13

“The Bitter Lesson”

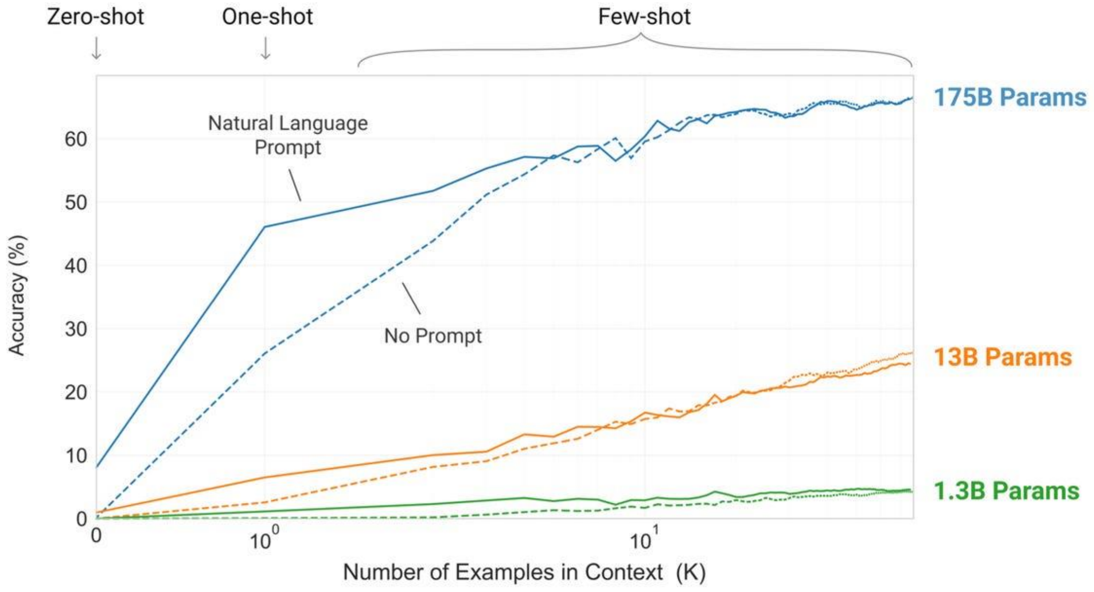
“The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation.” (Sutton, 2019)

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Emergence of new behaviors

Emergence means that the behavior of a system is implicitly induced rather than explicitly constructed. For Deep Learning, emergence is often induced by larger model & more data.

Example: Compared to GPT-2's 1.5B parameter parameter model, GPT-3's 175-billion model permits "prompting", i.e., adapting to a new task simply by describing task.

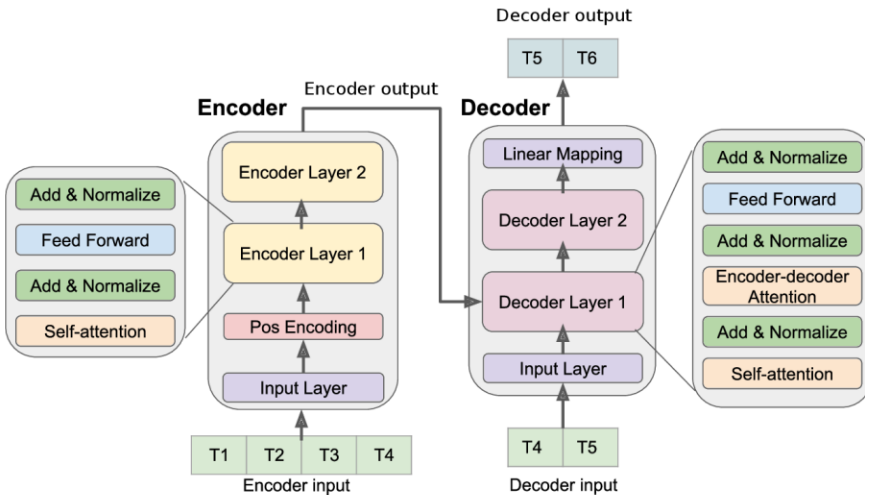


On the Opportunities and Risks of Foundation Models (Bommasani et al., 2021)

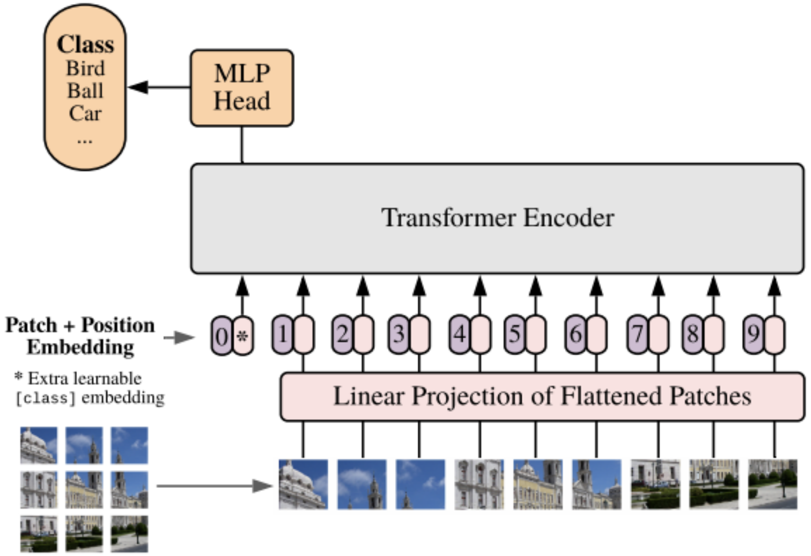
Homogenization of Architectures

Homogenization is the **consolidation** of methodologies for building machine learning systems across a wide range of applications.

Example: The Transformer Models (Vaswani *et al.,* 2017)



Transformer Models originally designed for NLP



Almost identical model (Visual Transformers) can be applied to Computer Vision tasks

On the Opportunities and Risks of Foundation Models (Bommasani et al., 2021)

Problems with Deep Learning

- Problem#1: Lack of a formal understanding
  - Non-Convex! Non-Convex! Non-Convex!
    - Depth>=3: most losses non-convex in parameters
  - Worse still, existing intuitions from classical statistical learning theory don’t seem to carry over.
  - Theoretically, we are stumbling in the dark here

- Standard response #1
  - “Yes, but this just means there’s new theory to be constructed”
  - “All interesting learning problems are non-convex”

- Standard response #2
  - “Yes, but it often works!”

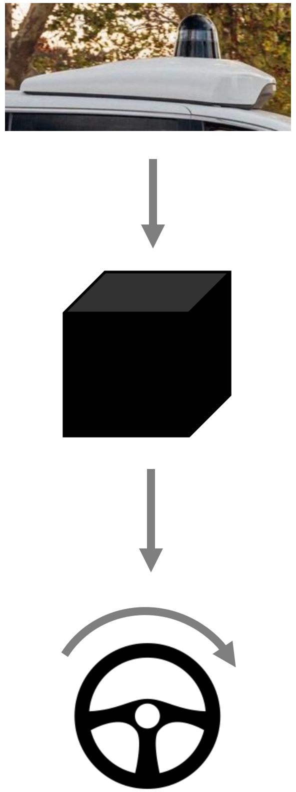
Problems with Deep Learning

- Problem#2: Lack of interpretability
  - Hard to track down what's failing
  - Pipeline systems have expected performances at each step
  - In end-to-end systems, it's hard to know why things are not working

Problems with Deep Learning

- Problem#2: Lack of interpretability

Why did the robot do that?



Problems with Deep Learning

- Problem#2: Lack of interpretability
  - Hard to track down what's failing
  - Pipeline systems have expected performances at each step
  - In end-to-end systems, it's hard to triage an error

- Standard response #1
  - Tricks of the trade: visualize features, add losses at different layers, pre-train to avoid degenerate initializations…
  - "MOOOORE DATA!"
  - "We're working on it"

- Standard response #2
  - "Yes, but it often works!"

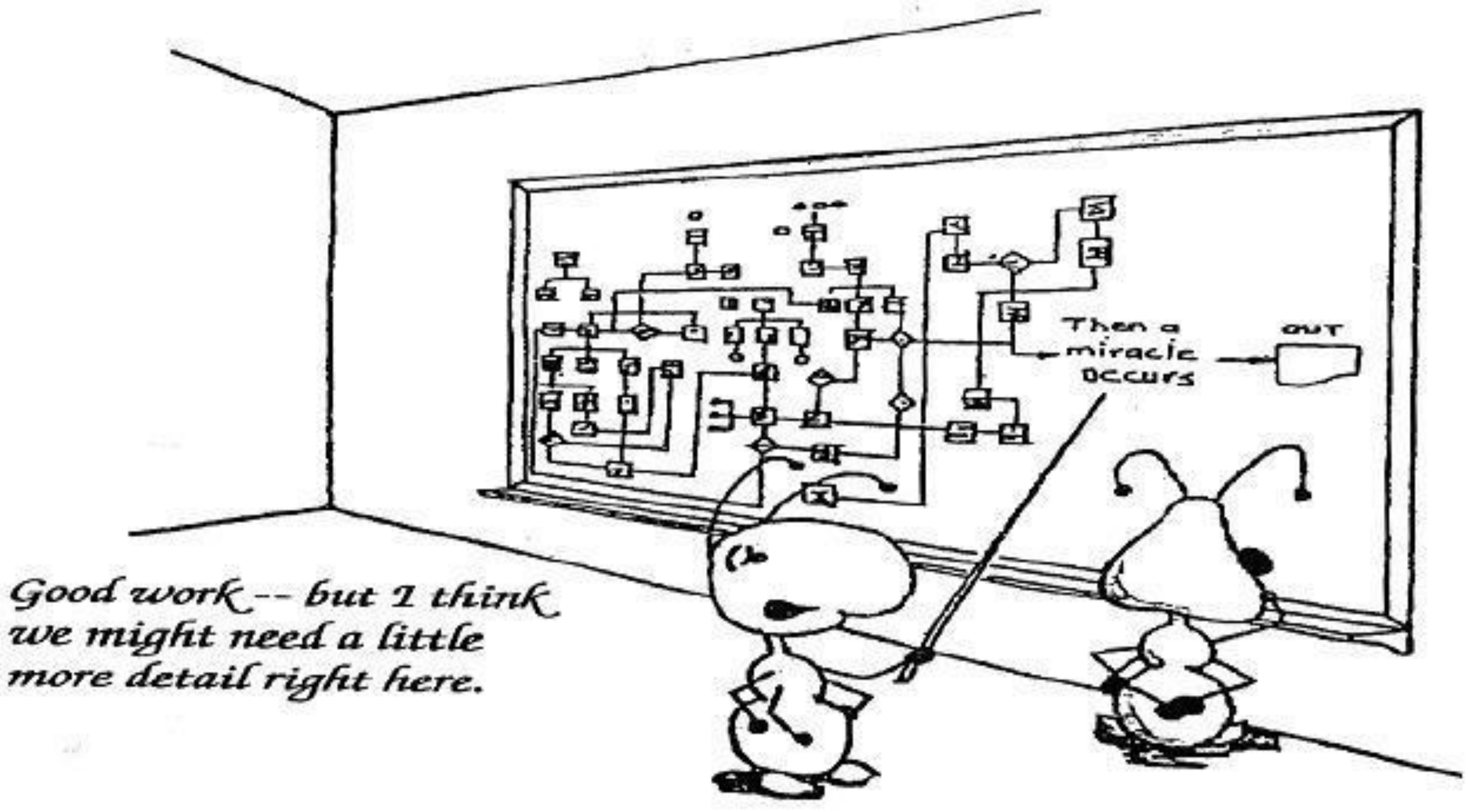
Problems with Deep Learning

- Problem#3: Lack of easy reproducibility
  - Direct consequence of stochasticity & non-convexity
    - different initializations → different local minima
  - Other stochasticity in the training pipeline: parallel data loading, distributed training, numerical precision on GPU…

- Standard response #1
  - It's getting much better
  - Standard toolkits/libraries/frameworks now available
  - PyTorch, TensorFlow, MxNet…

- Standard response #2
  - "Yes, but it often works!"

Yes it works, but how?



Outline

- What is Deep Learning, the field, about?
  - Highlight of some recent projects from my lab

- What is this class about?
  - What to expect?
  - Logistics

- FAQ

Outline

- What is Deep Learning, the field, about?
  - Highlight of some recent projects from my lab
- What is this class about?
  - What to expect?
  - Logistics
- FAQ

What is this class about?

- Introduction to Deep Learning

- Goal:
  - After finishing this class, you should be ready to get started on your first DL research / engineering project.
    - Backpropagation, optimization
    - Convolutional Neural Networks (image data)
    - Recurrent Neural Networks / Transformers (sequence data)
    - Generative Models (unsupervised learning)
    - Deep Reinforcement Learning (decision making)
    - (Glimpses of) cutting-edge research in CV, NLP, Robotics
  - Work on fun projects with your peers!

- Target Audience:
  - Senior undergrads, MS-(CS, ML, …), and new PhD students

What is this class about?

A *grossly simplified* timeline of deep learning

This course

… Pioneers	Foundations	Explosion	The Era of Foundation Models …
1940-1960s	1970-2000s	2010-2019	2020-???

Pioneers (1940-1960s)

- Artificial Neural Networks
- Hebbian theory (Hebb)
- Perceptron (Rosenblatt)
- Visual cortex cells (Hubel and Wiesel)
- …

Foundations (1970-2000s)

- ConvNets (Fukushima)
- Backpropagation in NNs (Remelhart, Williams, Hinton, LeCun, …)
- LSTM (Hochreiter and Schmidhuber)
- GPUs
- …

Explosion (2010-2019)

- ImageNet (Jia, Fei-Fei, …)
- AlexNet (Krizhevsky)
- The Cat Experiment (Google)
- Transformer
- Deep Generative Models
- AlphaGo
- Deep Reinforcement Learning
- More GPUs
- …

The Era of Foundation Models (2020-???)

- GPT Series (OpenAI)
- CLIP & DALL·E (OpenAI)
- SegmentAnything (Meta)
- Llama Series (Meta)
- Bard (Google)
- StableDiffusion
- …

What this class is NOT

- NOT the target audience:
  - Students without sufficient background knowledge (Python, linear algebra, calculus, basic probability & statistics)
  - Advanced grad-students already working in ML/DL areas
  - People looking for an in-depth understanding of a research area that uses deep learning (3D Vision, Large Language Models, Deep RL, etc.).
- NOT the goal:
  - Intro to Machine Learning / Optimization

Caveat

- This is an ADVANCED Machine Learning class
  - This should NOT be your first introduction to AI/ML
  - You will need a formal class; not just self-reading/coursera

Prerequisites

- Python Programming
  - Basic knowledge of numerical computations & tools (e.g., numpy)
  - You will write a lot of code!
- Intro Machine Learning
  - Classifiers, regressors, loss functions, MLE, MAP
- Linear Algebra
  - Matrix multiplication, eigenvalues, positive semi-definiteness…
- Calculus
  - Multi-variate gradients, hessians, jacobians…
- Must read (on W3 reading list): Matrix calculus for deep learning
  - https://explained.ai/matrix-calculus/index.html

Your Teaching Team

- Instructor: Prof. Danfei Xu
- Ph.D. (2021), Stanford University
- 2022-Now: Assistant Professor at Georgia Tech
- Research in Robot Learning (Robotics + Machine Learning)
- Office in Klaus 1314
- Running, cycling, cooking, wrangling with robots

Your Teaching Team

Head TA: David He

Kun-Lin Hsieh

Lawrence Zhu

Mengying Lin

Srikar Balusu

Woo Chul Shin

Yangcen Liu

Yixin Zhang

Office Hour

TA Office Hours:

- Virtual over zoom
- Check course website for OH slots and zoom links
- Start next week

Danfei’s Office Hours:

- In-person (Klaus 1314) or zoom
- No assignment (PS/HW) questions
- Lecture content / project ideas / administrative / career advice, …

Main channel of communication: Piazza

- HW and Project Assignments will be announced on Piazza
- Q&A: Check other questions before posting new ones
- Access through Canvas

HWs / Project Deliverables

- Announced on Piazza
- Submit on **Gradescope** (access through Canvas)

Organization & Deliverables

- 4 problem-sets+homeworks (64%)
  - Mix of theory (PS) and implementation (HW)
  - First one goes out next week
    - Start early, Start early, Start early, Start early, Start early, Start early

- Course project (36%)
  - Projects done in groups of 2-4
  - You need a good reason to do a solo project.
  - Proposal (1%), Milestone Report (10%), Final Report (20%), Poster Session (5%)
  - **Find a team ASAP! Talk to people, use Piazza “find a teammate” post.**
  - Ideas & scope: http://cs231n.stanford.edu/reports.html

- (Bonus) Class Participation (1%)
  - Top (endorsed) contributors on Piazza

Late days

- 2 late days (48 hours)
- Submissions within the late days will receive a 20% penalty.
- After late days period, you get a 0 (no excuses except medical)
  - Send all medical requests to dean of students (https://studentlife.gatech.edu/)
  - Form: https://gatech-advocate.symplicity.com/care_report/index.php/pid224342?

- **DO NOT SEND US ANY MEDICAL INFORMATION!** We do not need any details, just a confirmation from dean of students

PS0

- Out already. Due Friday Aug 22^{th}, 11:59pm
  - Will be available on class webpage
  - If not registered yet (on waitlist), see webpage FAQ for form to request gradescope access

- Grading
  - Not counted towards your final grade, but required
  - If it takes you more than 3 hours to complete, you might struggle in the course.

- Topics
  - PS: probability, calculus
  - HW: Numpy, calculus

Class Project

- Goal
  - Chance to try Deep Learning in practice
  - Encouraged to apply to your research (computer vision, NLP, robotics, compbio,…)
  - Must be done this semester.
  - Can combine with other classes, but **separate thrust**
    - get permission from both instructors; delineate contribution to each course
  - 2-4 members (outside of this requires approval)
  - Will have a separate lecture on this in Week 4
  - UG and Grad students are allowed to team up

Computing

- Major bottleneck
  - GPUs

- Options
  - Google Colab Pro (free for students!)
    - jupyter-notebook + free GPU instance
  - PACE ICE (GT's instruction cluster)
    - Details will be sent later
  - Your own / group / advisor's resources

4644 vs 7643

- Level differentiation

- Separate grade curves calculation
  - As a result, 4644 and 7643 may have different letter grade cut-offs.

Outline

- What is Deep Learning, the field, about?
  - Highlight of some recent projects from my lab
- What is this class about?
  - What to expect?
  - Logistics
- FAQ

Waitlist / Audit / Sit in

- Waitlist
  - Class is full.
  - Do PS0 **NOW**. Come to first few classes.
  - Hope people drop. Big churn in the first week.

- "I need this class to graduate"
  - Talk to your degree program advisor. They control the process of making sure you have options to graduate on time.

- Audit or Pass/Fail
  - No.

What is the re-grading policy?

- Homework assignments
  - **Within 1 week** of receiving grades: submit regrade request on GradeScope

What is the collaboration policy?

- Collaboration
  - Only on HW (coding) and project.
  - You may discuss the questions
  - Each student writes their own answers
  - Write on your homework anyone with whom you collaborate
  - Each student must write their own code for the programming part

- Zero tolerance on plagiarism
  - Neither ethical nor in your best interest
  - Always credit your sources
  - Don’t cheat. We will find out.

How do I get in touch?

- Primary means of communication -- Piazza
  - No direct emails to Instructor **unless private information**
  - Instructor/TAs can provide answers to everyone on forum
  - Class participation credit for answering questions!
  - No posting answers. We will monitor.
  - Stay respectful and professional.

Share your feedback

Ways to share your feedback:

- Come talk to us
- Email
- Private Piazza Post
- **Anonymous feedback form (linked on Piazza)**

Questions?