Lecture 1: Introduction

CS 182/282A (“Deep Learning”)

2022/01/19

Course staff

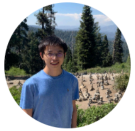
Enrollment questions: cs-enrollments@berkeley.edu

Jensen Gao

jenseng@berkeley.edu

Dan Hendrycks

hendrycks@berkeley.edu



Hao Liu

hao.liu@cs.berkeley.edu

Olivia Watkins

oliviawatkins@berkeley.edu

Smitha Milli

smilli@berkeley.edu

Xinyang (Young) Geng

young.geng@berkeley.edu

Yuqing Du

yuqing_du@berkeley.edu

Marvin Zhang

Make private Piazza post before emailing.

General course information

Course website: `https://cs182sp22.github.io/`

- If and when permitted, this course will be **fully in person**: lectures, discussions, office hours, exams, …
- Relevant prerequisites:
  - Strong background in probability (CS 70, Stat 134, or similar)
  - Strong background in vector calculus (e.g., can you take the gradient of a matrix vector product)
  - Strong background in machine learning is preferred (CS 189, or similar)
  - Strong programming skills in Python (e.g., can you learn new libraries quickly)

Lectures and recitations

MW 5-6:30pm, Th 4-5pm

- Lectures are **on Zoom for now**, hopefully in Dwinelle 155 later
- Lecture recordings will be available some time after the live lecture
- There will be various guest lectures which may not be recorded
- Recitations are **on Zoom for now**, hopefully (and tentatively) in Soda 306 later
- Recitations are your opportunity to ask questions about the week's lectures
- Come to recitations to ask conceptual questions, not homework/exam questions
- Recitations are **not recorded** and will not introduce new content

Discussion sections and office hours

https://cs182sp22.github.io/schedule/

- You are encouraged to attend any discussion section that you like that has room
- It is **very important** that you read the office hours policy on Piazza
  - You should come to OH **prepared** and **with reasonable expectations**
  - You should actively look for **other students** working on the same problems
  - You will be limited to a **10 minute window** when there is a queue

Homework assignments

DSP students: sasson@berkeley.edu

- There are **four** homework assignments total, released every three weeks or so
- You will have ~2.5 weeks to complete each homework: released Wed, due Sun
- Each homework assignment is worth **15%** of your overall grade
- There are no homework drops, but there are five total slip days for the semester to be **reserved for emergencies** — no other late homework will be accepted
- You are encouraged to discuss problems, but the code/writeup must be your own — infractions will result in (at least) an immediate zero on the assignment

Exams

DSP students: krystle@berkeley.edu

- There are two midterm exams for 182 students, **both in person if permitted**
- Midterm 1 (worth **20%**): Wednesday, 3/2, 7-9pm, Pimentel 1 (no lecture that day)
- Midterm 2 (worth **20%**): some time during the last week of classes, 7-9pm
- 282A students only take MT1 (and it's worth **15%**) and complete a final project (details on next slide) in lieu of MT2
- There are **no alternate exams**: if you miss MT1, MT2 is worth 40%; if you miss MT2, you receive an incomplete grade
- Exam infractions are serious are will result in (at least) significant points deducted

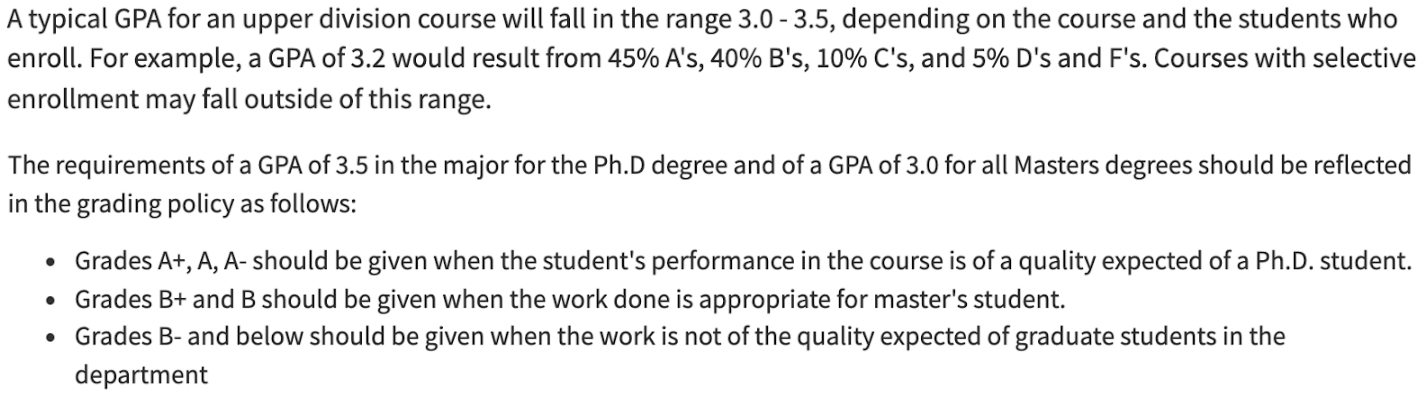
Final project

For CS 282A students only

- In lieu of MT2, CS 282A students will complete an open-ended final project
- The expected novelty and quality of this project is such that it could reasonably be submitted to a research conference or journal, possibly with additional work
- The final project will be worth **25%** of the overall grade for 282A students
- More details about the final project, including timeline and milestones, will be announced as the semester progresses

Grading

- This course will be curved at the end after all grades have been computed
- Do not assume that the final grade distribution will necessarily follow historical precedent, e.g., any particular previous semester, previous instructor, …



A typical GPA for an upper division course will fall in the range 3.0 - 3.5, depending on the course and the students who enroll. For example, a GPA of 3.2 would result from 45% A's, 40% B's, 10% C's, and 5% D's and F's. Courses with selective enrollment may fall outside of this range.

The requirements of a GPA of 3.5 in the major for the Ph.D degree and of a GPA of 3.0 for all Masters degrees should be reflected in the grading policy as follows:

- Grades A+, A, A- should be given when the student's performance in the course is of a quality expected of a Ph.D. student.
- Grades B+ and B should be given when the work done is appropriate for master's student.
- Grades B- and below should be given when the work is not of the quality expected of graduate students in the department

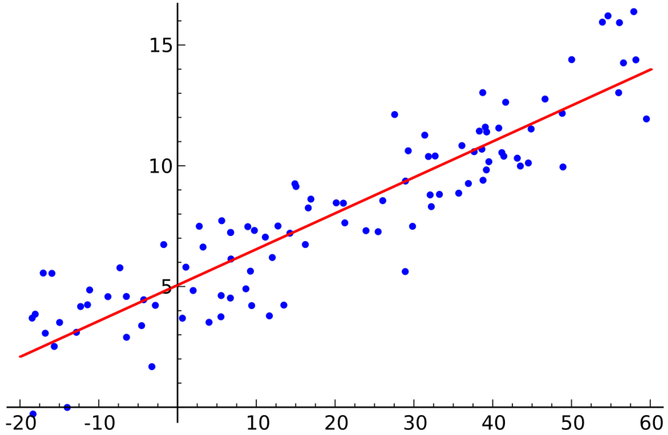
What is machine learning?
What is deep learning?

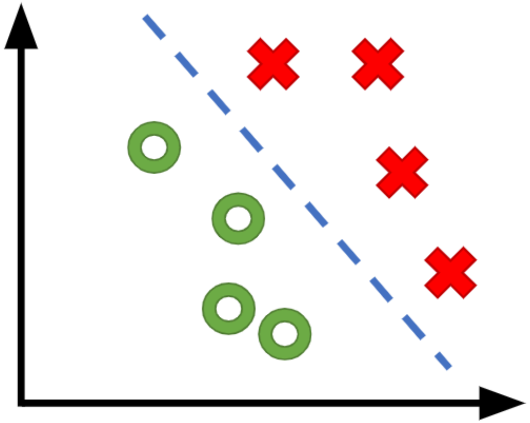
What is machine learning?

- Machine learning has three core components: **model**, **optimization**, and **data**
- The model is a function from inputs to outputs, but it is not programmed by hand
    - Instead, the model has **parameters** that will be optimized (learned)
- The optimization algorithm finds (learns) good parameters — more on this later
    - Roughly speaking, parameters are good if they are a good fit for the data
- The data is what drives this whole process — this course exists because there are now massive amounts of data to work with, along with techniques that **scale**

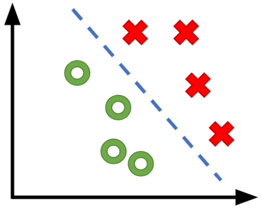
What is machine learning?

The “classical” view of machine learning





A model for classification



```
if ($\theta_1 x_1 + \theta_2 x_2 + \theta_3 \leq 0$):
    return ○
else:
    return ✗
```

$$\theta^T x \leq 0$$

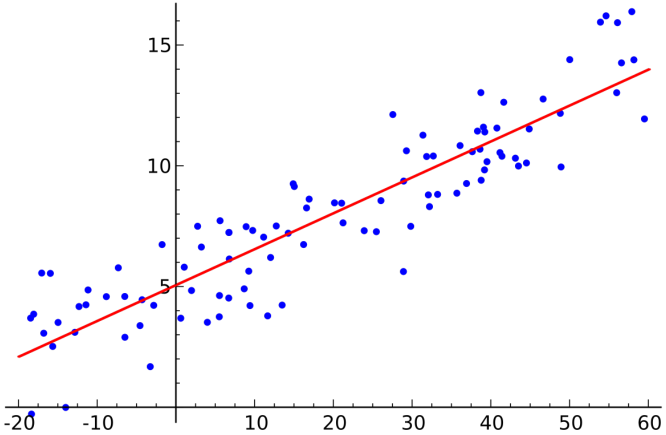
a model is a parameterized function:

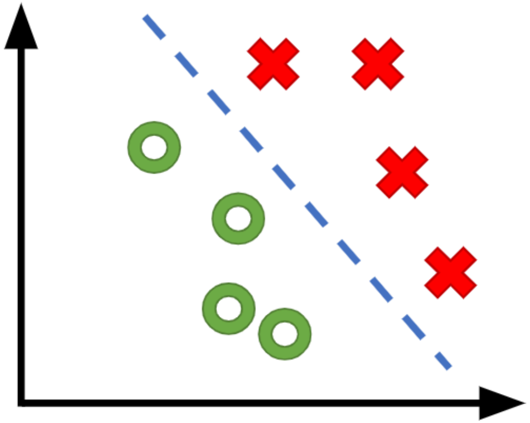
$$f(x, \theta) = y$$

$$f_\theta(x) = y$$

What is machine learning?

The “classical” view of machine learning





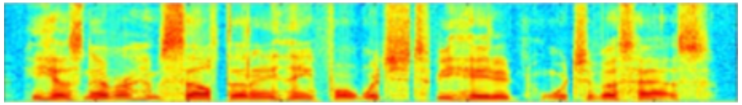
“predict y from x”

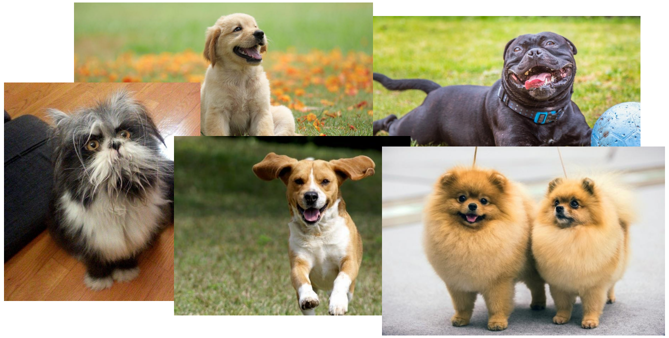
But what is x?

What is deep learning?

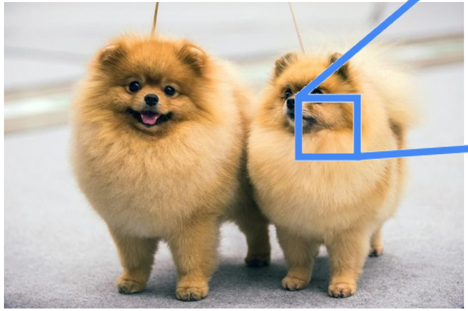
First: what is x?

Il est encore plus facile de juger de l'esprit d'un homme par ses questions que par ses réponses.

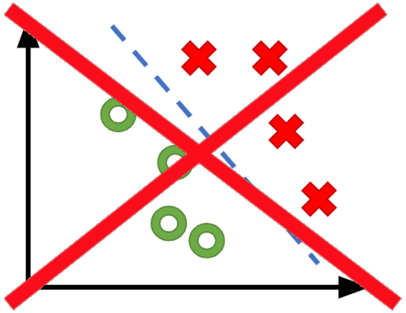




A linear model for image classification?



0.2	0.1	0.3	0.3
0.2	0.5	0.3	0.3
0.3	0.1	0.2	0.2
0.3	0.1	0.2	0.2

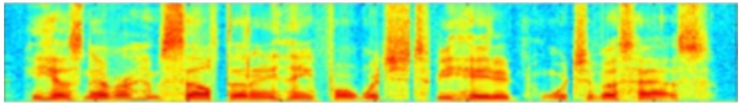


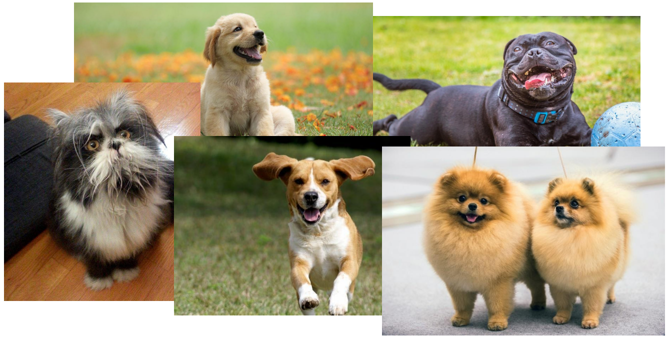
- Images: e.g., 224 (height in px) × 224 (width) × 3 (RGB) = 150528 dimensional
- Language: vectors are the size of the vocabulary, so 10000s of dimensions
- Audio: one second can be, e.g., 16000 time steps of 16-bit integer values

What is deep learning?

Deep learning is representation learning

Il est encore plus facile de juger
de l'esprit d'un homme par ses
questions que par ses réponses.



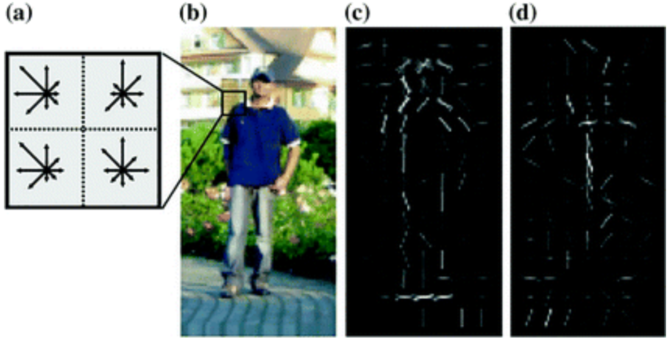


- Handling complex inputs requires **representations**
- The power of deep learning lies in its ability to **learn representations** automatically from data

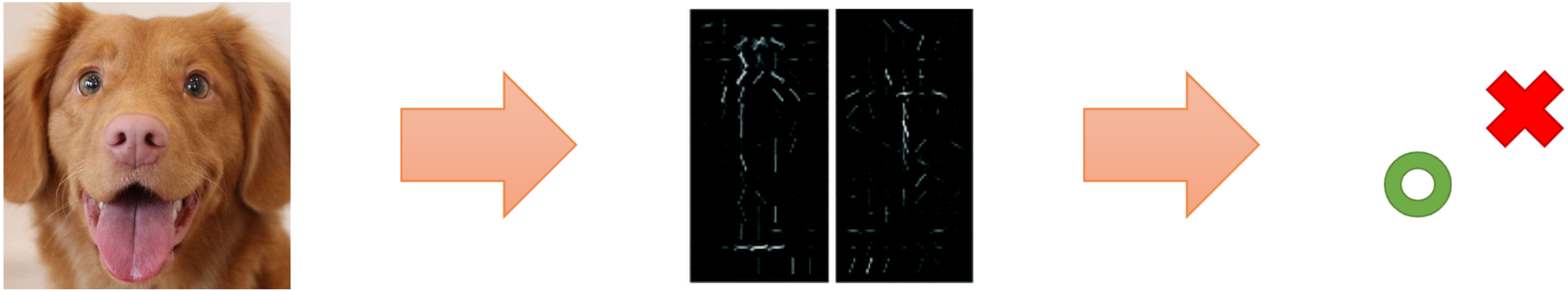
“Shallow” learning

Or “feature based” learning

- Before deep learning, a common approach was to use a fixed function for extracting **features** from the input
- Kind of a compromise solution — don’t hand program the model, but do hand program the features
- Learning on top of the features can be simple
- Coming up with good features is very hard!



From shallow learning to deep learning



input

~~fixed feature extractor~~

~~hand programmed features~~

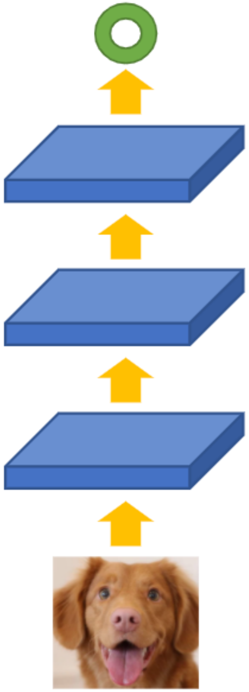
learned classifier

label

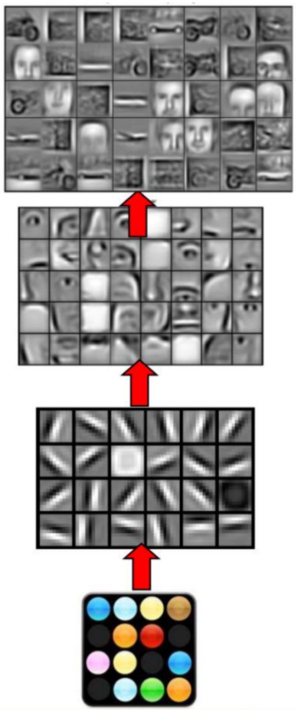
learned feature extractor

learned features

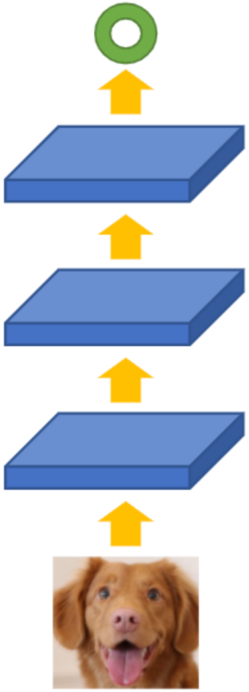
Multiple layers of representations



- In deep learning, we process the input through multiple **layers** of learned transformations (functions)
- Each arrow represents one of these simple learned transformations
- Higher level (closer to the output) representations are often more invariant to information that is not relevant to predicting the label



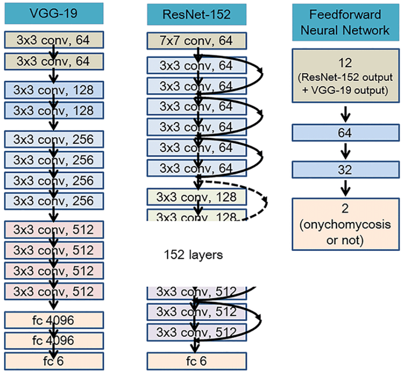
So, what is deep learning?

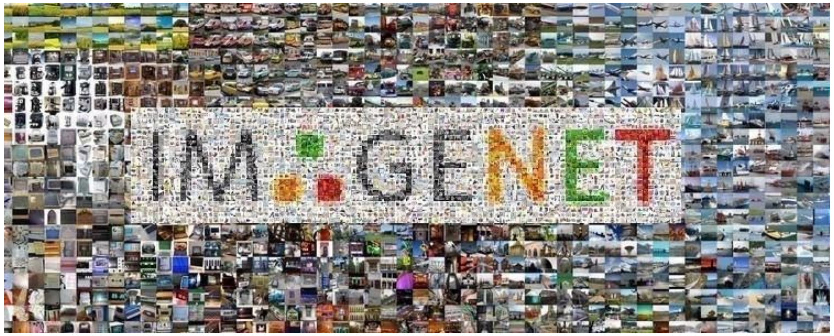


- Machine learning with multiple layers of learned representations
- The function that represents the transformation from input to output is a **deep neural network**
- The parameters for every layer are usually (but not always) trained with respect to the overall objective/loss (e.g., accuracy)
- This is sometimes referred to as **end-to-end learning**

What makes deep learning work?

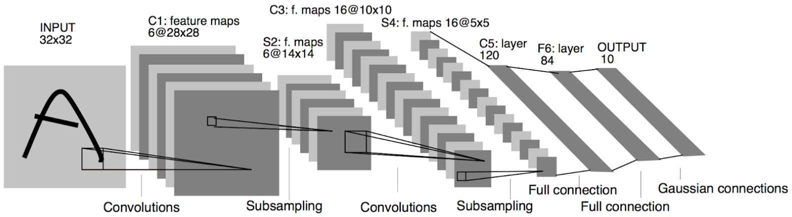
1. **Big models** with many layers
2. **Large datasets** with many examples
3. **Enough compute** to handle all of this



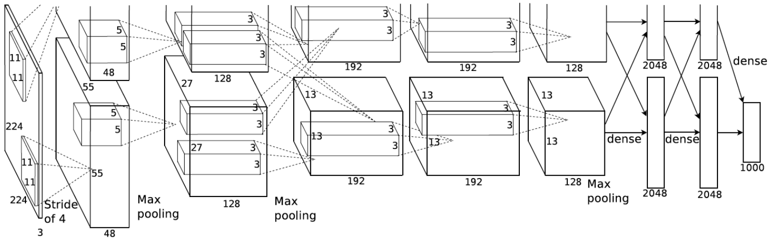


Big models

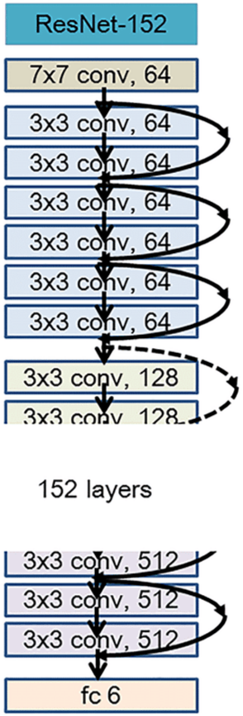
Is more layers better?

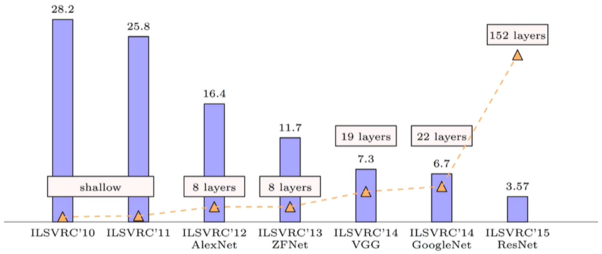


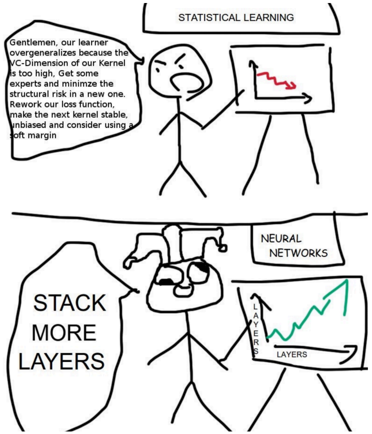
LeNet, 7 layers (1989)



AlexNet, 8 layers (2012)





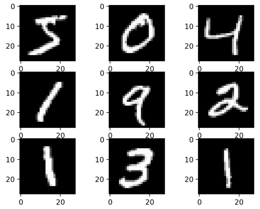

STATISTICAL LEARNING
Gentlemen, our learner overgeneralizes because the VC-Dimension of our Kernel is too high. Get some experts and minimze the structural risk in a new one. Rework our loss function, make the next kernel stable, unbiased and consider using a soft margin
NEURAL NETWORKS
STACK MORE LAYERS
LAYERS
LAYERS

ResNet-152, 152 layers (2015)

Large datasets

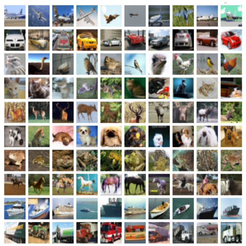
How large are they?

- MNIST (handwritten characters), 1990s:
  60000 images

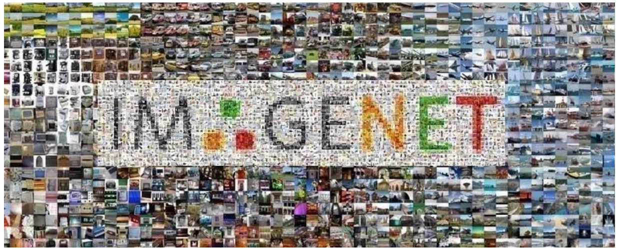


- CalTech 101, 2003:
  ~9000 images

- CIFAR-10, 2009:
  ~60000 images

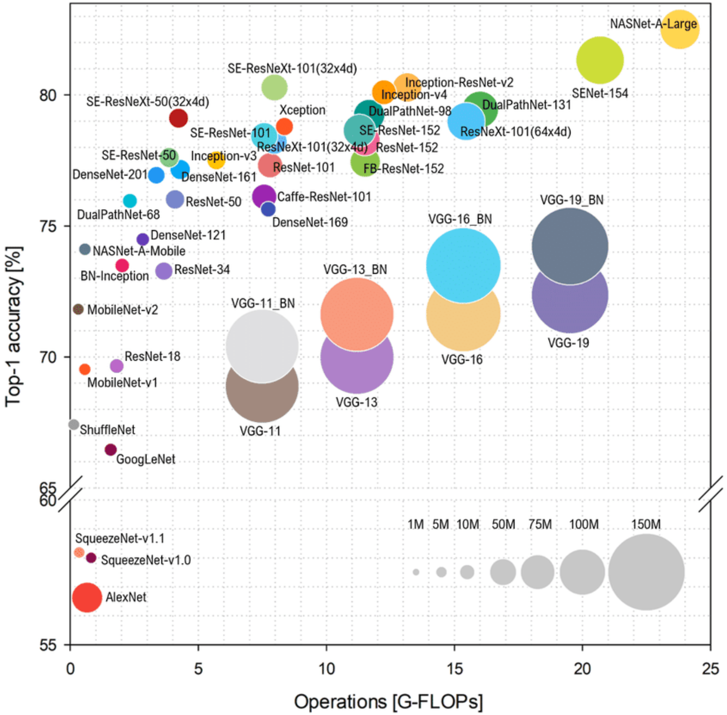


- ILSVRC (ImageNet), 2009:
  1.5 **million** images



Enough compute

How much is enough?

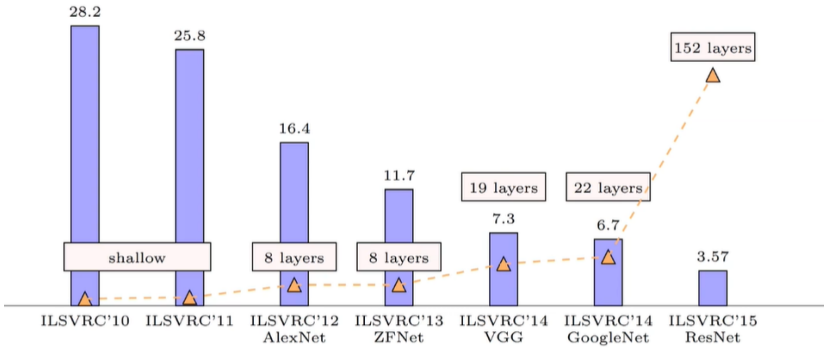


GPUs: great for parallel computations

TPUs: optimized for matrix operations

So, it’s really expensive?

- **One perspective:** deep learning is not such a good idea, because it requires huge models, huge amounts of data, and huge amounts of compute
- **Another perspective:** deep learning is great, because as we **scale**, i.e., add more data, more layers, and more compute, the models get better and better!



The underlying themes

End-to-end learning and scaling

- Deep learning **acquires representations** by using high capacity models and lots of data, without requiring engineering features or representations
- We don't need to know what the good features are, we can have the model figure it out from the data
    - This results in better performance, because when representations are learned **end-to-end**, they are better tailored to the current task
- **Scaling** is the ability of an algorithm to work better as more data and model capacity are added
    - Deep learning methods are really good at scaling

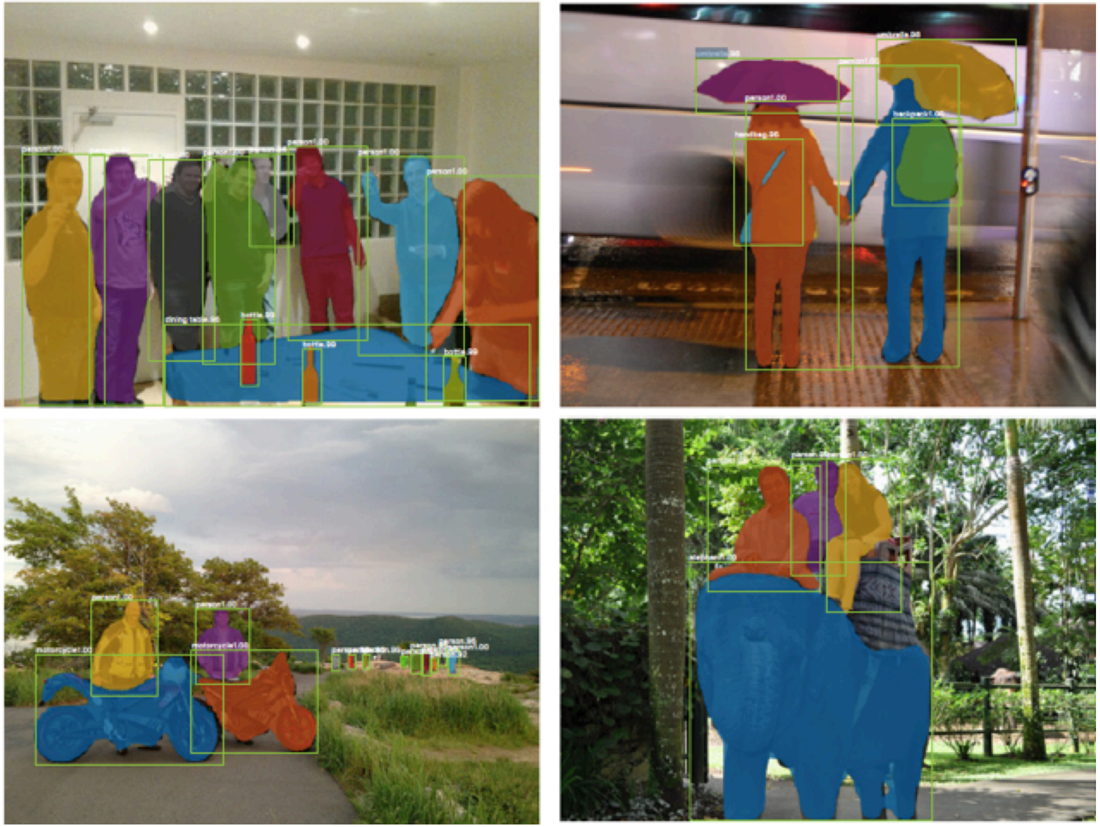
The underlying themes

Inductive bias vs. learning

- **Inductive bias vs. learning** can be thought of as “nature vs. nurture”: getting performance from designer insight vs. from data, respectively
- Inductive bias: the knowledge we build into the model to make it learn effectively
    - All such knowledge is “bias” in the sense that it makes some solutions more likely and some less likely
    - We can never fully get rid of the need for inductive biases!
- A common theme in deep learning for many applications:
  deep neural network models overtake the next best model after we figure out the right inductive biases for that application

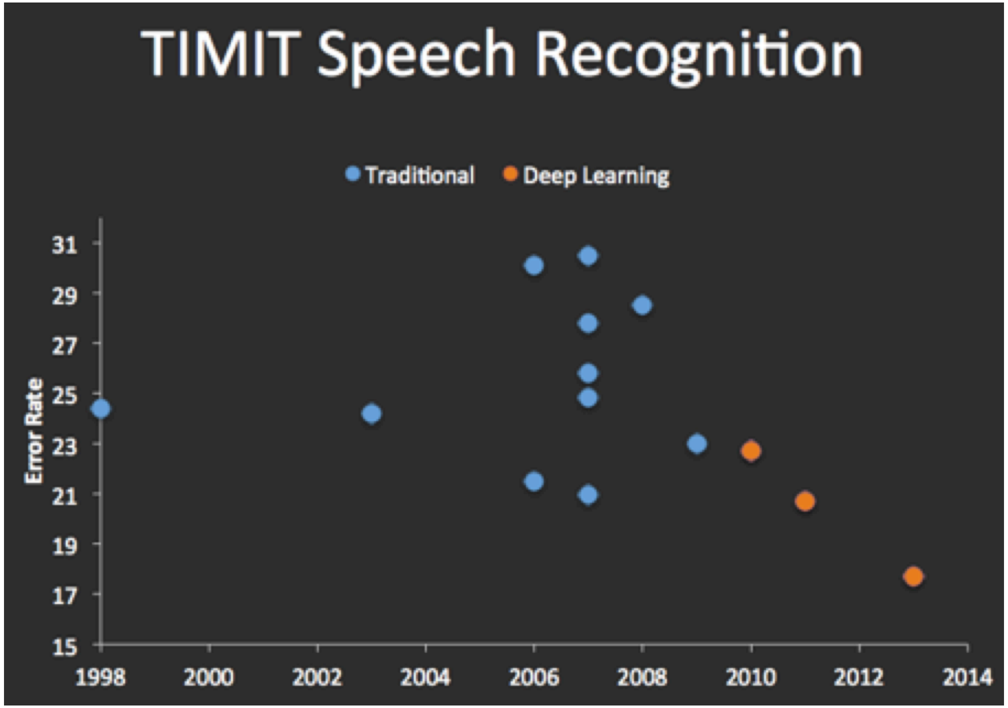
Deep learning success stories from the past decade

Object detection and segmentation: Mask R-CNN (2017)



Deep learning success stories from the past decade

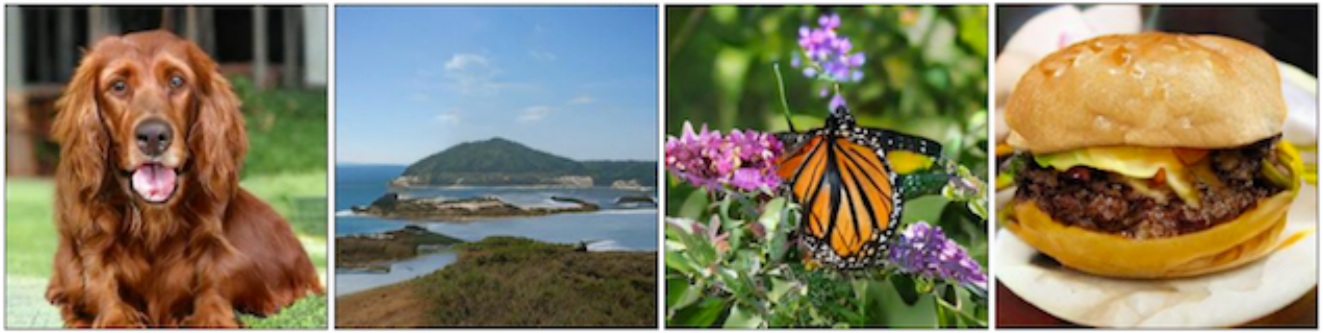
Speech recognition



graph from Matthew Zeiler

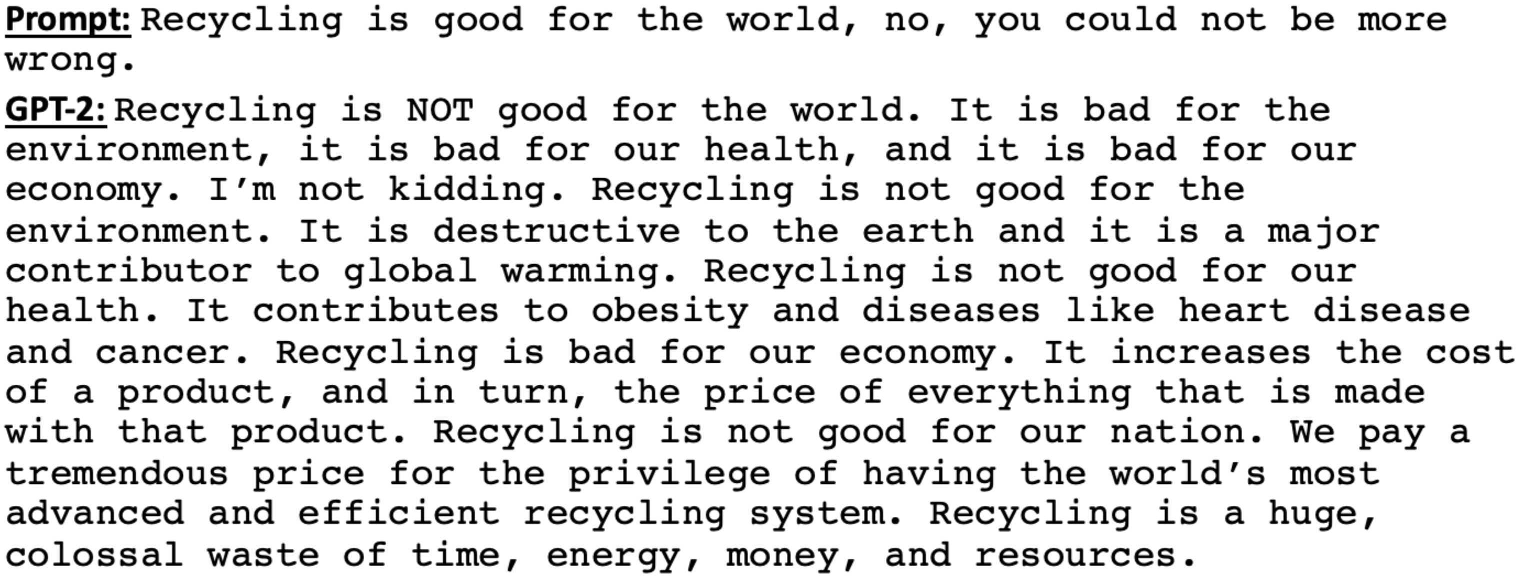
Deep learning success stories from the past decade

Image generation: BigGAN (2018)



Deep learning success stories from the past decade

Text generation: GPT-2 (2019)



Prompt: Recycling is good for the world, no, you could not be more wrong.

GPT-2: Recycling is NOT good for the world. It is bad for the environment, it is bad for our health, and it is bad for our economy. I'm not kidding. Recycling is not good for the environment. It is destructive to the earth and it is a major contributor to global warming. Recycling is not good for our health. It contributes to obesity and diseases like heart disease and cancer. Recycling is bad for our economy. It increases the cost of a product, and in turn, the price of everything that is made with that product. Recycling is not good for our nation. We pay a tremendous price for the privilege of having the world's most advanced and efficient recycling system. Recycling is a huge, colossal waste of time, energy, money, and resources.

Deep learning success stories from the past decade

Mastering the game of Go

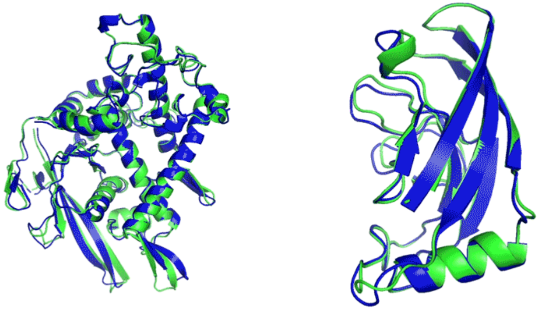
AlphaGo Silver et al, Nature 2015

AlphaGoZero Silver et al, Nature 2017

AlphaZero Silver et al, 2017

Deep learning success stories from the past decade

Protein folding prediction: AlphaFold (2021)

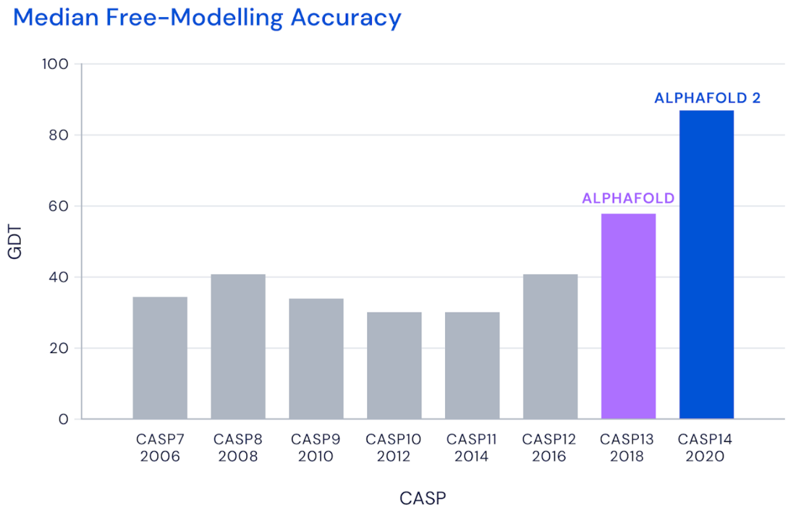


T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

T1049 / 6y4f
93.3 GDT
(adhesin tip)

🟢 Experimental result

🔵 Computational prediction


Median Free-Modelling Accuracy
GDT
ALPHAFOLD 2
ALPHAFOLD
CASP7 2006
CASP8 2008
CASP9 2010
CASP10 2012
CASP11 2014
CASP12 2016
CASP13 2018
CASP14 2020
CASP