COMP 432 Machine Learning

Introduction

Computer Science & Software Engineering
Concordia University, Fall 2021

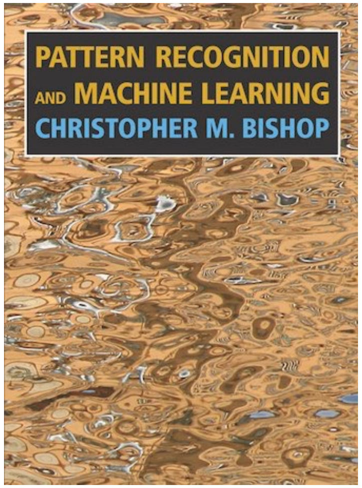
Preliminaries

Course web page

- Moodle only; find syllabus there

Textbook

- Pattern Recognition and Machine Learning by Christopher M. Bishop



Marking

- 10% Labs (1% each, best 10 of 11)
- 24% Assignment
- 24% Quizzes (6% each, best 4 of 5)
- 42% Project
- no exams

Preliminaries

- Instructor:
  - Andrew Delong, Assistant Professor
- Prerequisites
  - Linear algebra
  - Multivariable calculus
  - Probability and statistics (basic)
  - Python programming skills
- Policies
  - Communicate with staff using Moodle
  - Read all policies in syllabus (lab, quiz, project, etc)

Goals of the course

- Understand how ML algorithms work
  - Theory: motivations, math, concepts
  - Practice: code from scratch + use libraries
  - Research: propose your own ideas
- Become a solid ML practitioner
  - Preprocess data, train models, evaluate
- Ace an ML job interview
  - Fluent in reading/writing/talking about ML

Goals of this lecture

1. What is machine learning about?
    - How is it related to traditional software?
    - How is it different from AI?
    - Examples
2. Linear models
    - Linear regression / minimizing errors
    - Directly solving versus gradient descent
    - Logistic regression / binary classification
    - High-dimensional features / feature normalization

What is machine learning *about*?

Machine learning

From Wikipedia, the free encyclopedia

Machine learning (**ML**) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.

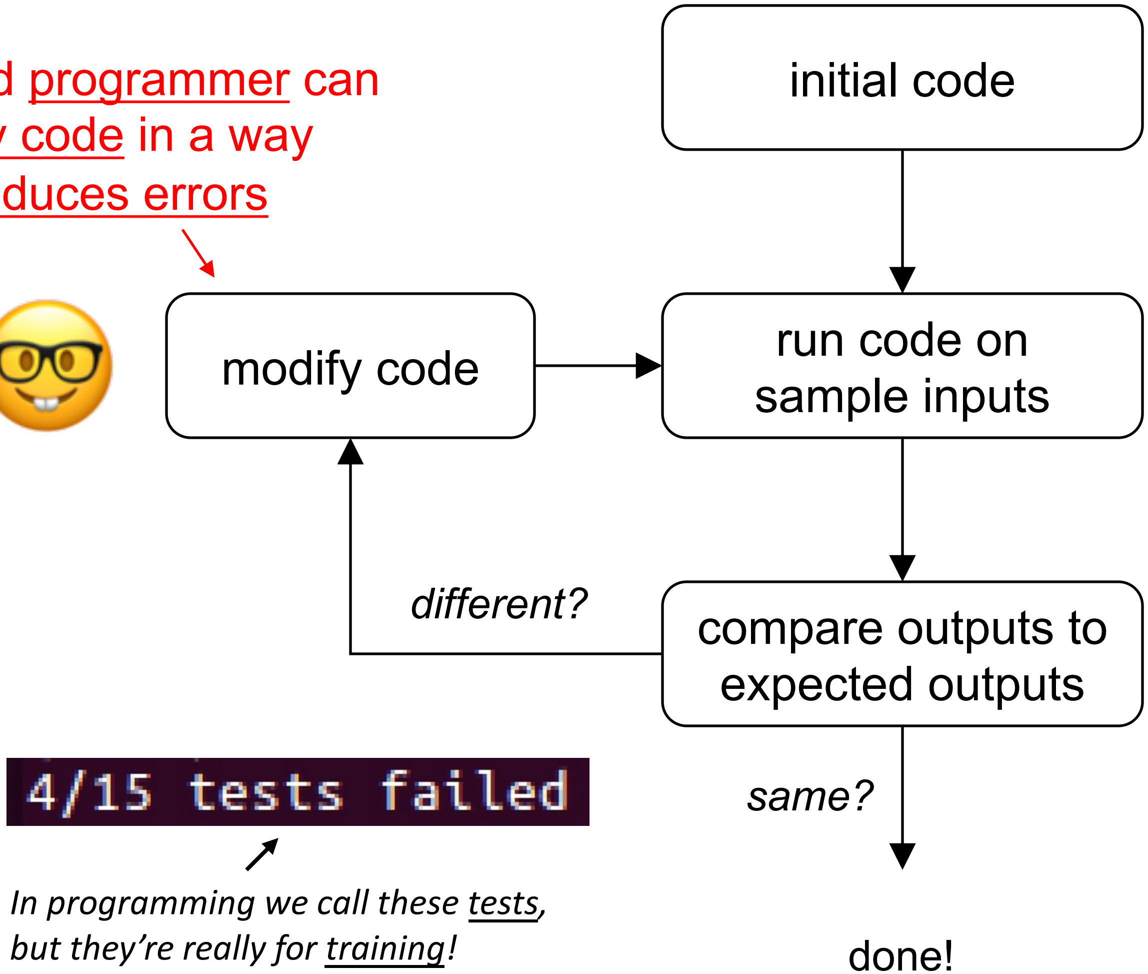
Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

How is machine learning *different* from programming?

How is it *similar* to programming?

Compare test-driven programming...

A good <u>programmer</u> can <u>modify code</u> in a way that <u>reduces errors</u>

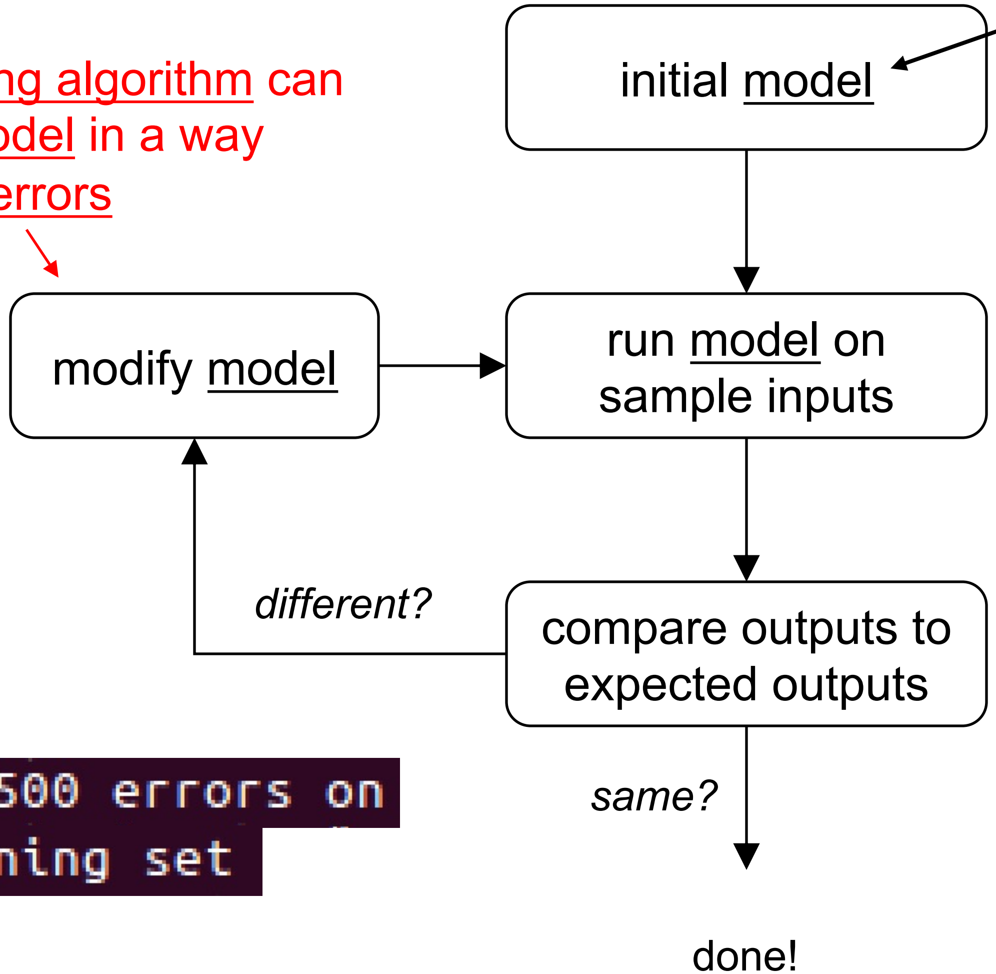


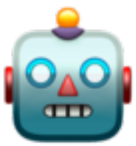
In programming we call these <u>tests</u>, but they're really for <u>training</u>!

… to supervised machine learning

A good learning algorithm can modify the model in a way that reduces errors

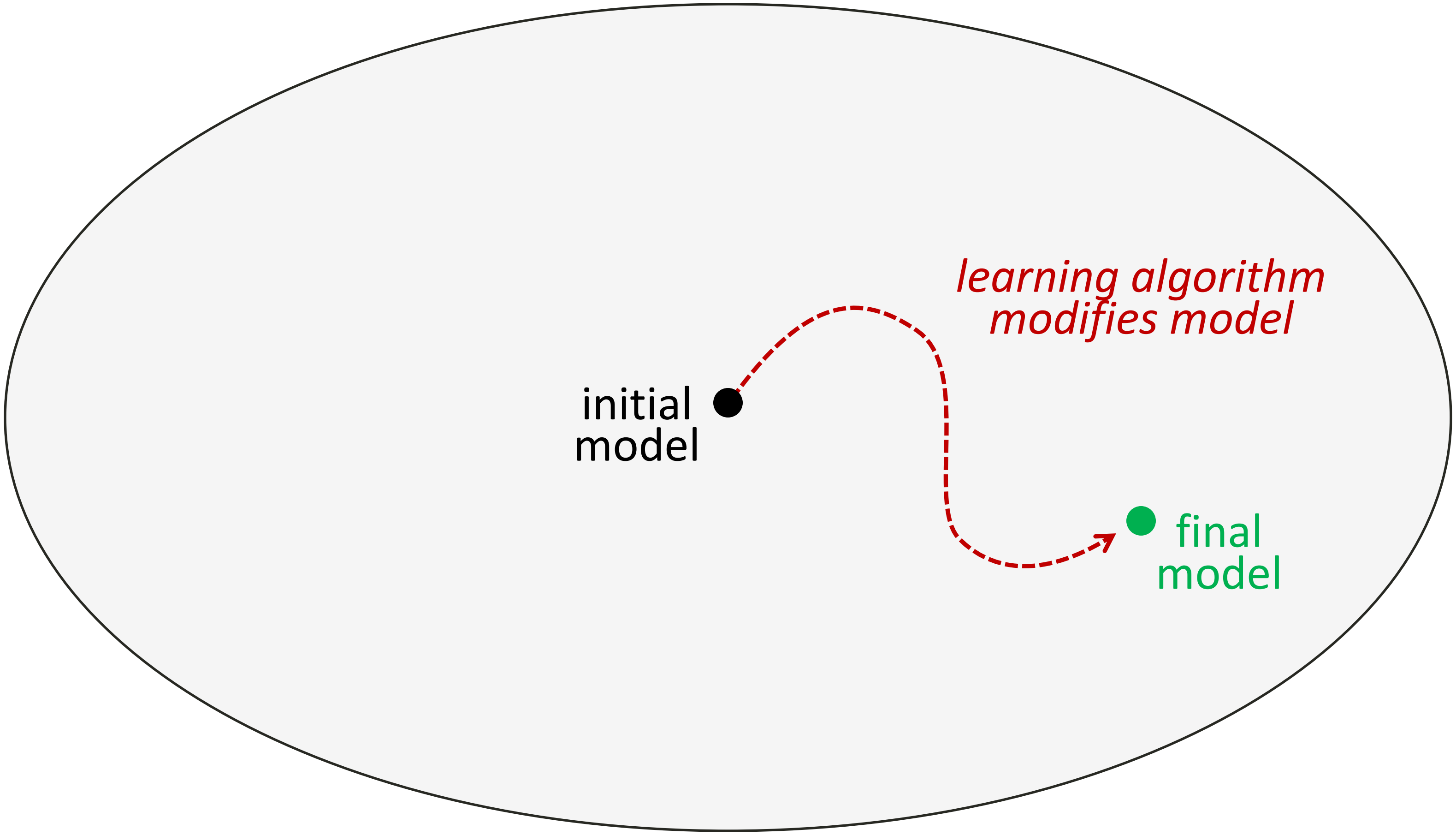
Think of a "model" as a specific class of programs that are easy to "tune" algorithmically





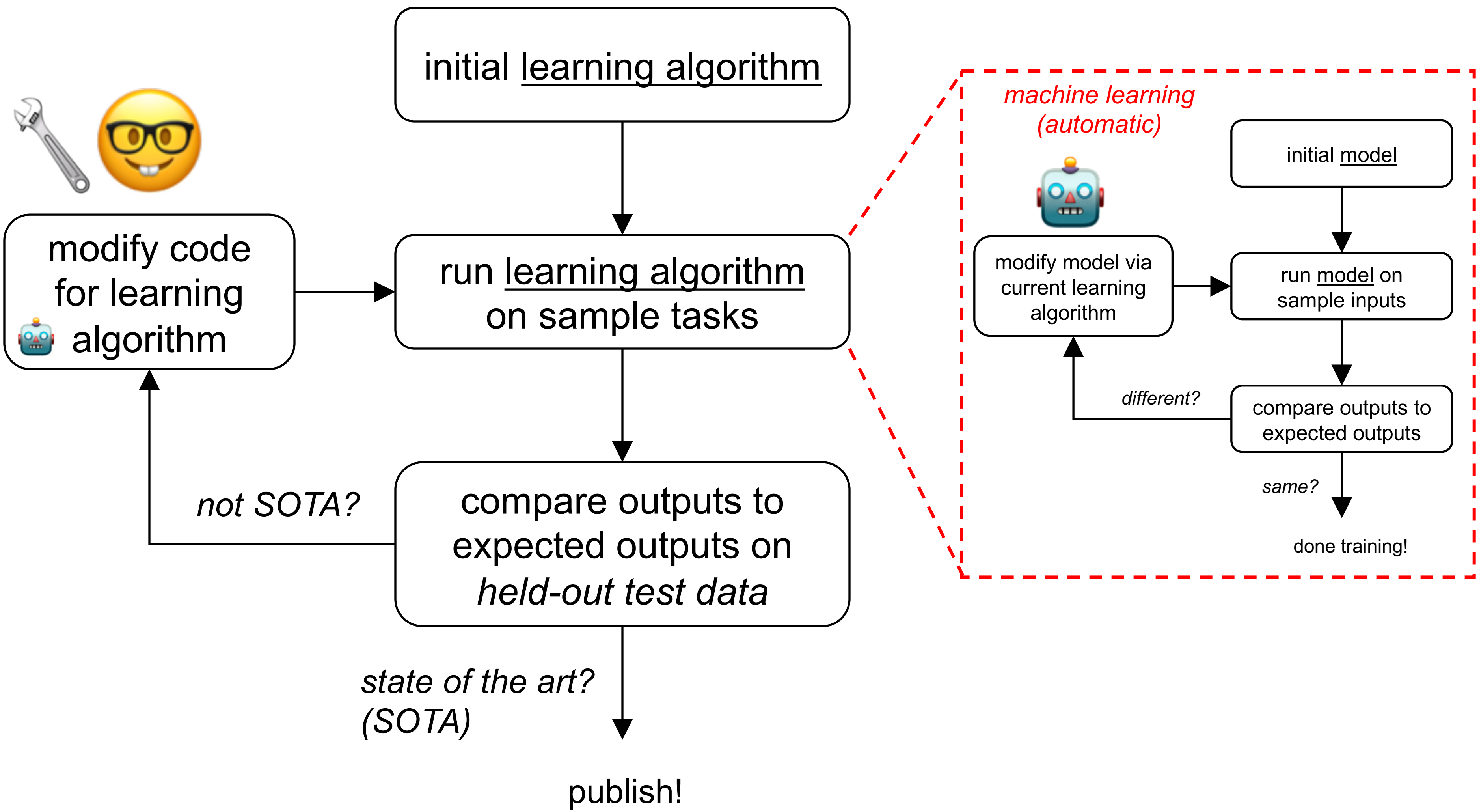
```
42/1500 errors on training set
```

… to supervised machine learning



… or to machine learning *research*

Research is test-driven programming of a machine learning algorithm!



Why is it exciting for society that machine learning *works*?

Because it opens new frontiers of <u>automation</u> 🤖

Machine learning systems better than humans at many important tasks

Yann LeCun
2018 Turing Award winner
Inventor of ConvNets

Yann LeCun
@ylecun

Recent progress in AI have taught us an important fact: an intelligence as general as human intelligence applied to a particular task totally sucks compared to an AI system dedicated to that task.
(also, human intelligence is not that general).

2:15 PM · Jul 17, 2019 · Facebook

Yann LeCun
@ylecun

Replying to @valentinp

Actually, recent progress in AI show that humans suck at designing AI systems.
That why we have to rely on machine learning to "design" them for us.

5:53 PM · Jul 17, 2019 · Twitter for Android

“The real AI revolution is that AI is a new way to build software.”

“For the last 40 years we have programmed computers.

For the next 40 years, we will train them.”

- Christopher M. Bishop, in 2020

Applications of machine learning

- Image search
- Object detection
- Fraud detection
- Agriculture optimization
- Financial forecasting
- Speech-to-text
- Text-to-speech
- Control systems for robotics
- Autonomous vehicles
- Autonomous drones
- Loan approval
- Medical diagnosis
- Algorithm heuristics
- Genome analysis
- Protein engineering
- Video game AI
- Photo/video/audio manipulation
- Music generation
- Virus/trojan detection
- ...

Image credit: *The Singularity is Near* by Kurzweil

But, much of ML is still "curve fitting"

Judea Pearl
2011 Turing Award winner

Pearl: "As much as I look into what's being done with deep learning, I see they're all stuck there on the level of associations. Curve fitting. That sounds like sacrilege, to say that all the impressive achievements of deep learning amount to just fitting a curve to data. [But] it's still a curve-fitting exercise, albeit complex and nontrivial."

Interviewer: *The way you talk about curve fitting, it sounds like you're not very impressed with machine learning.*

Pearl: "No, I'm very impressed, because we did not expect that so many problems could be solved by pure curve fitting. It turns out they can."

Machine learning versus AI

- **Machine learning** is about *automatically* designing computer programs from *examples* of the desired program behaviour. Useful when we *don't know how* to write such a program by hand.
- **Artificial intelligence** is about creating computer programs that mimic human cognitive abilities: reasoning, planning, dialogue, finding analogies, self-directed learning, etc.
- **AI <u>relies</u> on ML,** because we have *examples* of human cognitive abilities yet we *don't know how* to write such programs. ML does this *automatically*.

Major categories of ML

Supervised learning. Given (input, output) examples, learn a function that predicts correct outputs on new inputs.

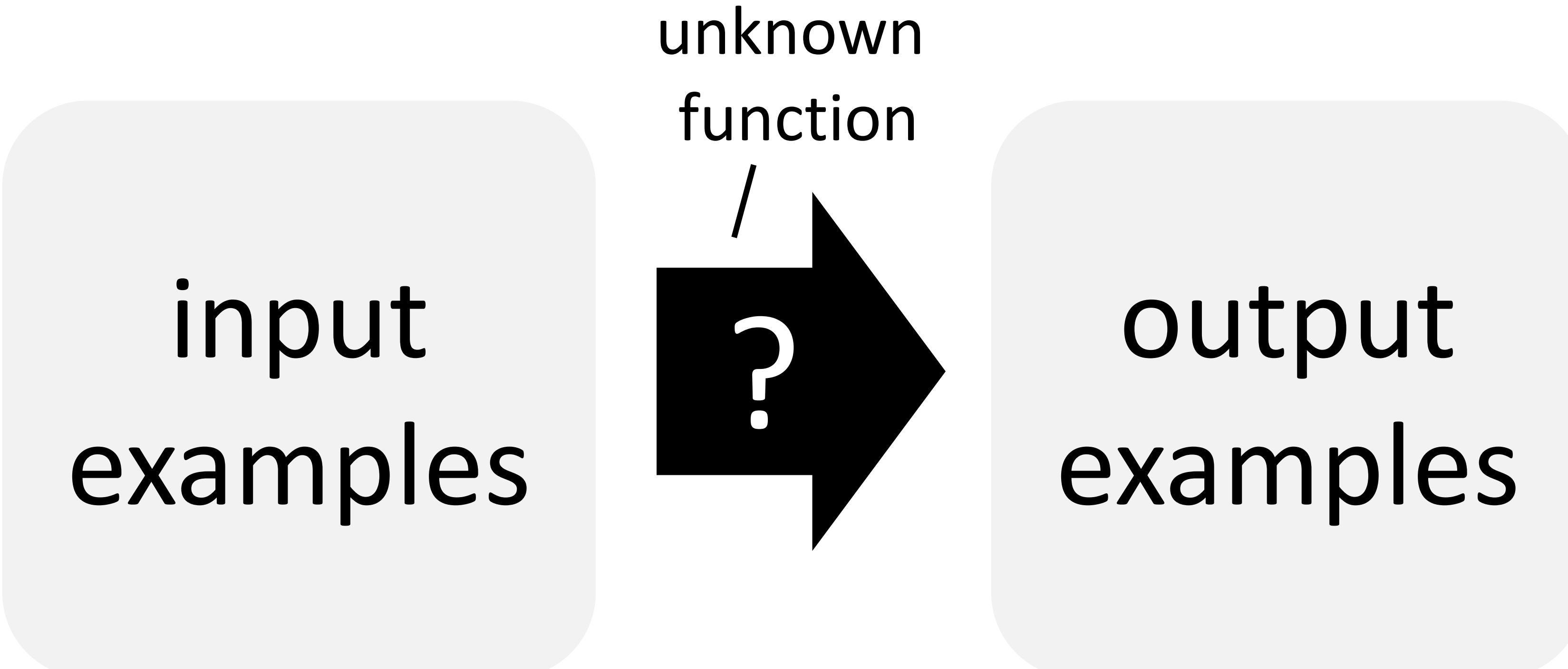
Semi-supervised learning is "between."
Transfer learning is conceptually related.

Unsupervised learning. Given input examples *only*, discover 'structure' in the data that are likely to be useful for some prediction task(s) or hand-analysis.

Reinforcement learning. Given an action space and a reward signal, learn a policy that predicts the action that will eventually maximize reward.

Examples of *supervised* learning (focus of today)

What is *supervised* learning?



Speech → Text


?
3G
9:41 AM
“On May 19 remind me it’s dad’s birthday”
Here’s your reminder for May 19, 2012 at 9 am:
19
Saturday
May 2012
Dad’s birthday
5/19/12
Cancel
Confirm

Language A → Language B

English

An admitting privilege is the right of a doctor to admit a patient to a hospital

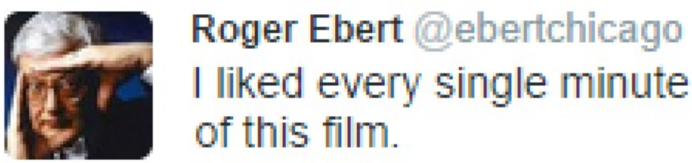
?

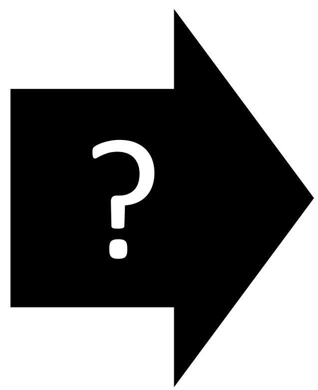
French

Un privilège d'admission est le droit d'un médecin de reconnaître un patient à l'hôpital

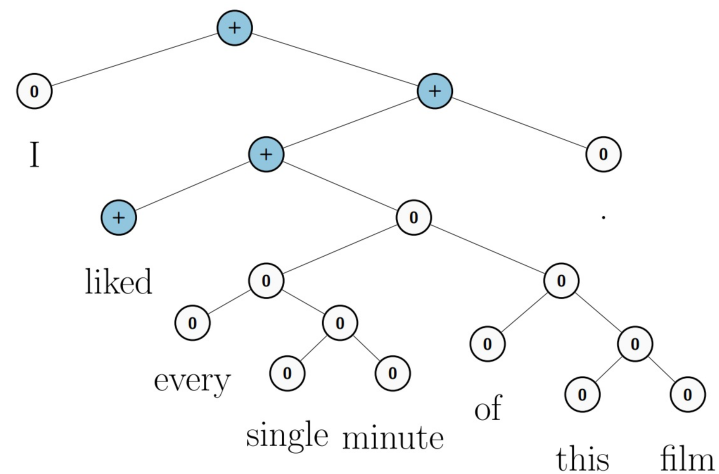
Neural Machine Translation by Jointly Learning to Align and Translate
Bahdanau et al., arXiv:1409.0473 2014.

Text → Meaning





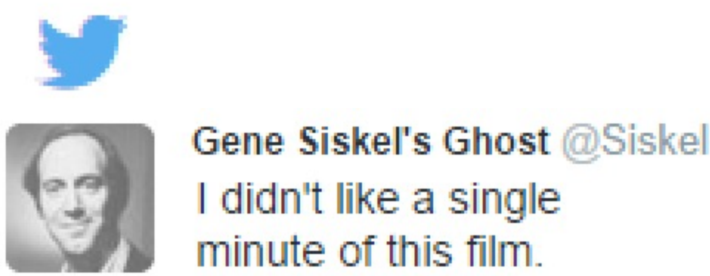
positive

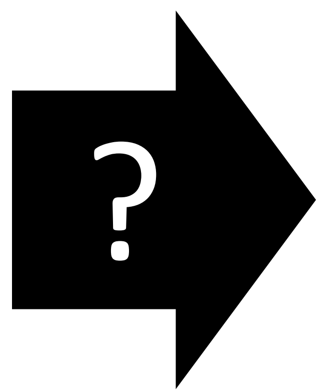


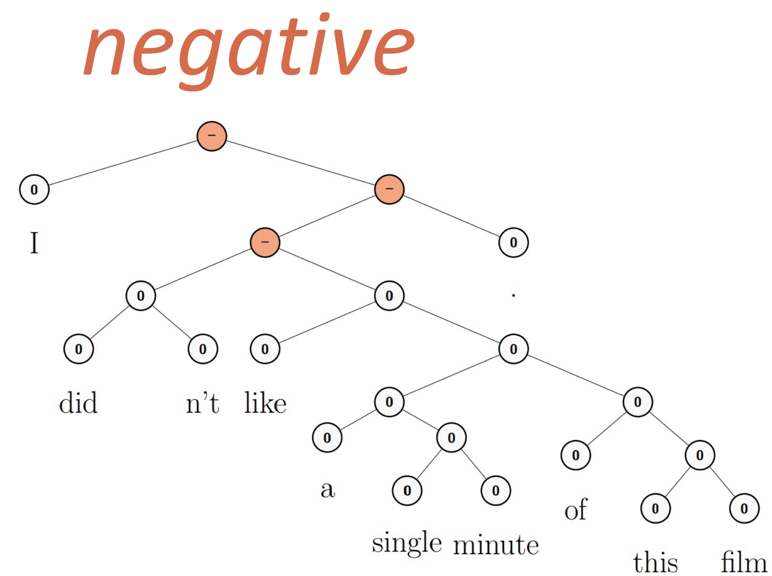
a structured output!

http://nlp.stanford.edu/sentiment/

Text → Meaning







http://nlp.stanford.edu/sentiment/

a structured output!

Image → Tags

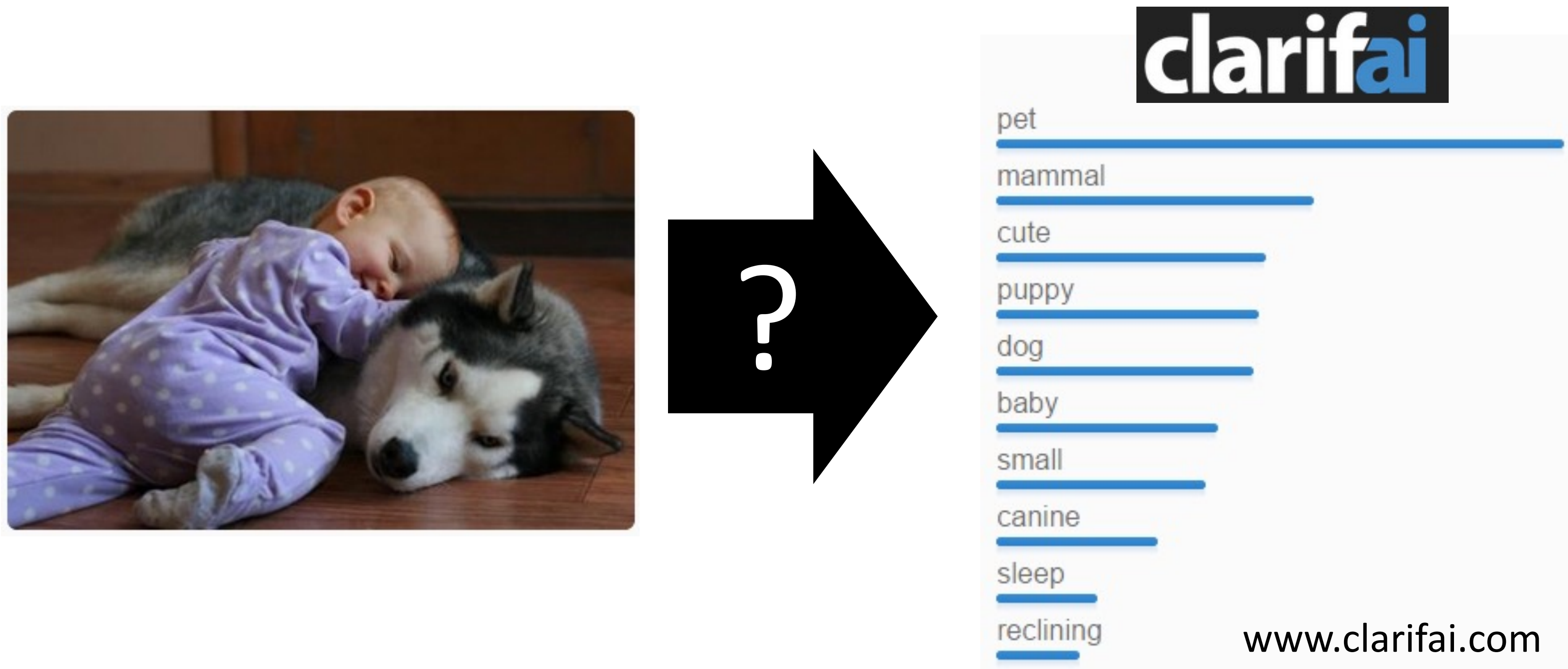


image tagging

Image → Tags

unknown function / ?

image tagging

The brain's "unknown function"

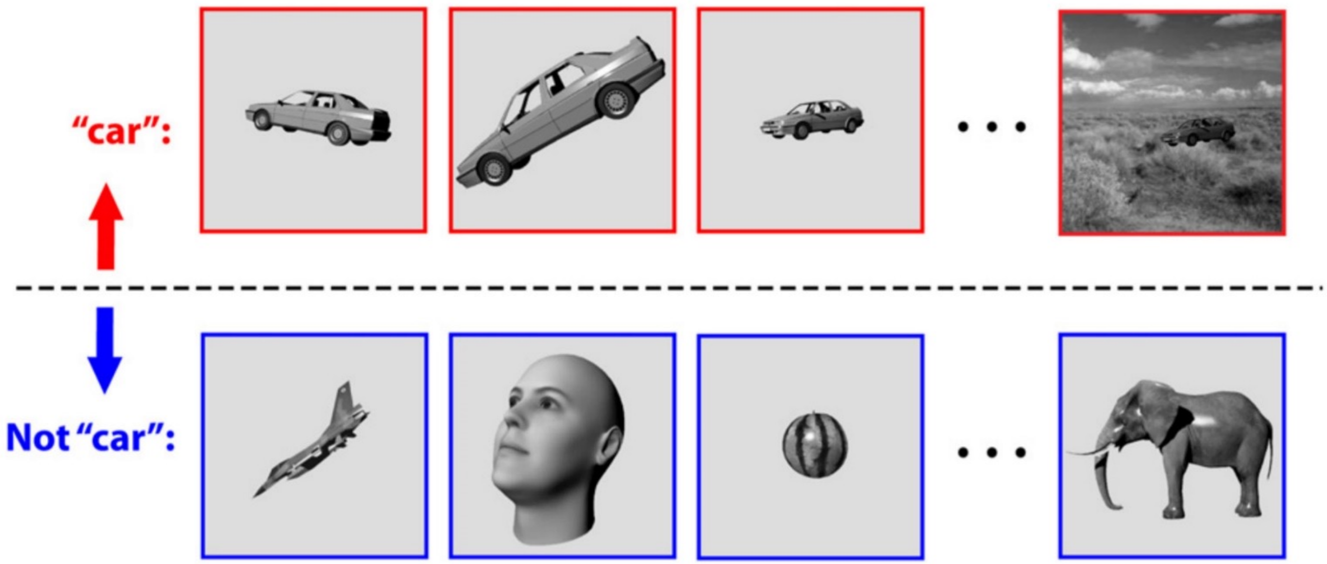


Figure 1. Core Object Recognition
Core object recognition is the ability to rapidly (<200 ms viewing duration) discriminate a given visual object (e.g., a car, top row) from all other possible visual objects (e.g., bottom row) without any object-specific or location-specific pre-cuing (e.g., DiCarlo and Cox, 2007). Primates perform this task remarkably well, even in the face of identity-preserving transformations (e.g., changes in object position, size, viewpoint, and visual context).

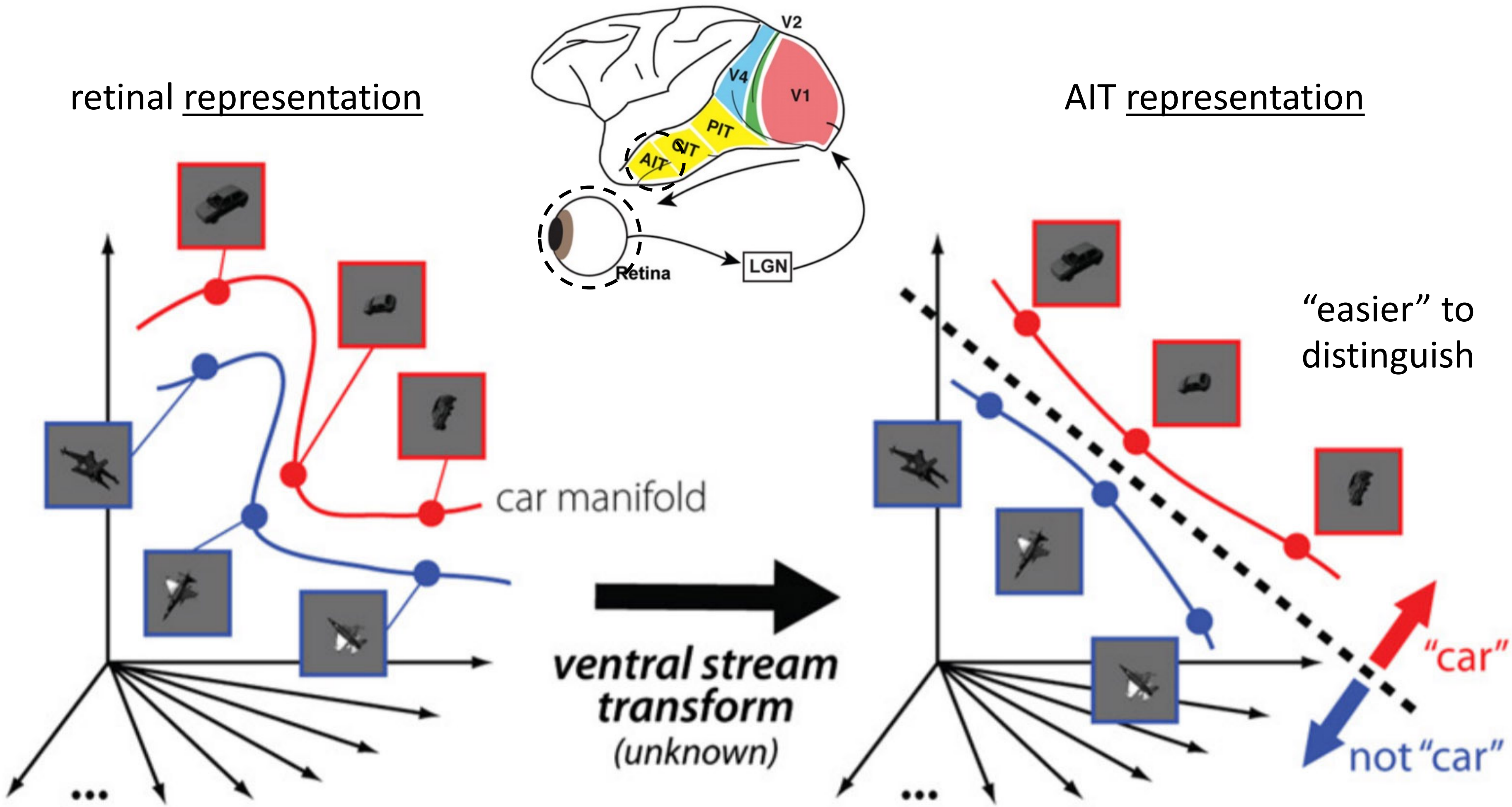
DiCarlo et al, *Neuron*, 2012

The brain’s “unknown function”

retinal representation

AIT representation

“easier” to distinguish

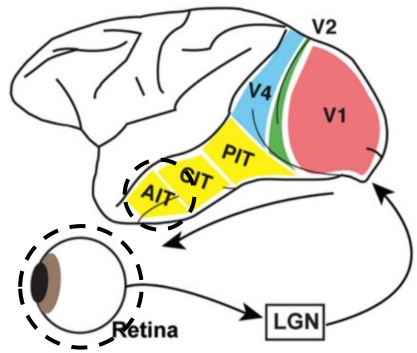


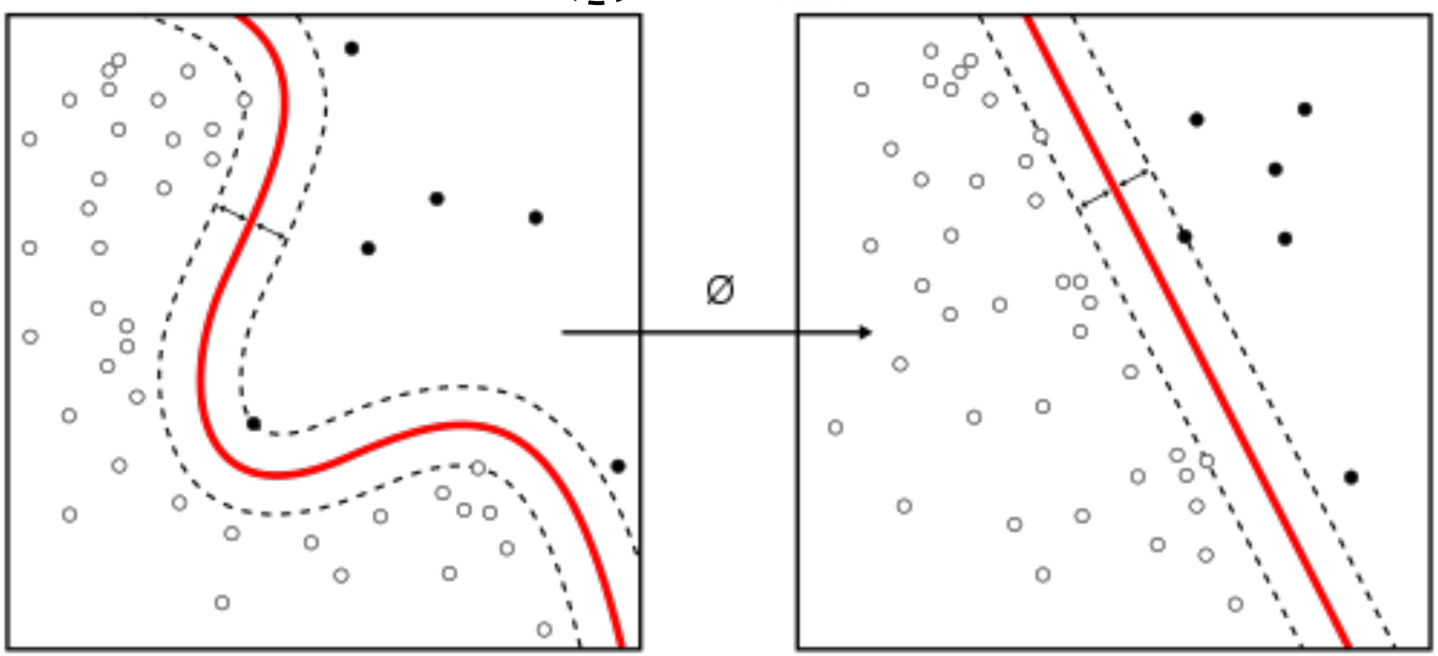
DiCarlo et al, *Neuron*, 2012

The brain’s “unknown function”

retinal representation

AIT representation

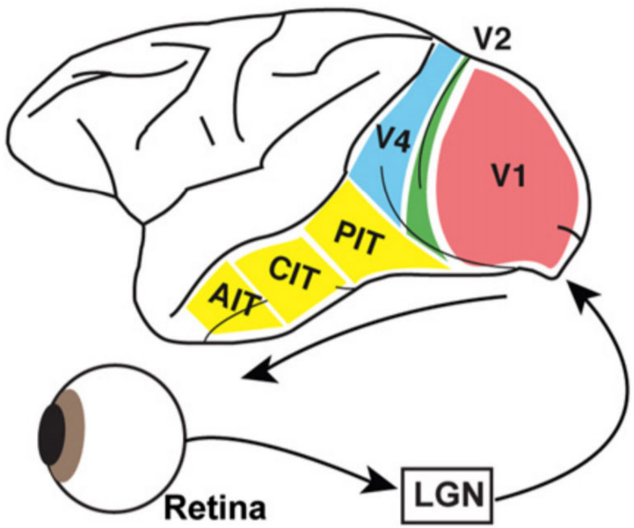


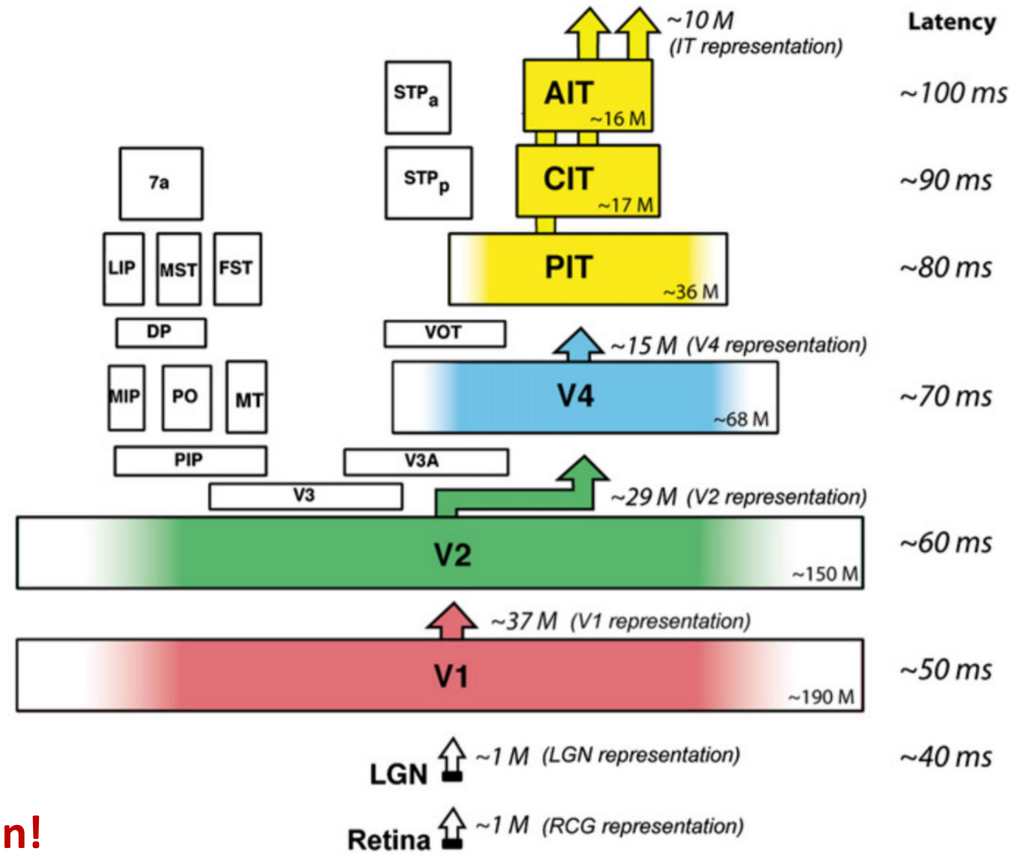


“easier” to distinguish

DiCarlo et al, *Neuron*, 2012

The brain’s “unknown function”





Somewhat “deep,” but also very *wide:*
LOTS of parallel “computation” going on!

Figure 3. The Ventral Visual Pathway
Ventral stream cortical area locations in the macaque monkey brain, and flow of visual information from the retina.

DiCarlo et al, *Neuron*, 2012

Image → Pixel Tags

high-dimensional output

http://www.clement.farabet.net/

Image → Caption

?

"A pizza sitting
on top of a pan
on top of a stove."

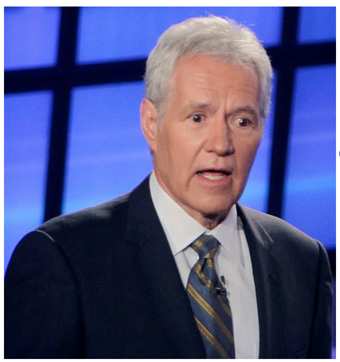
The New York Times

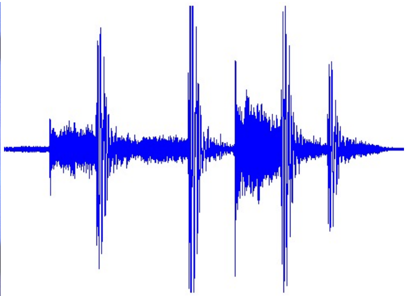
Nov 2014

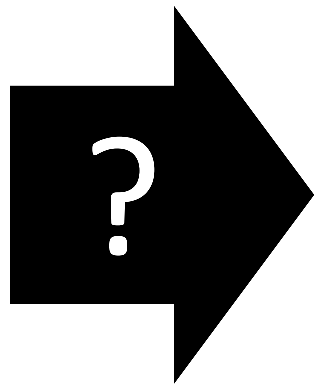
Google

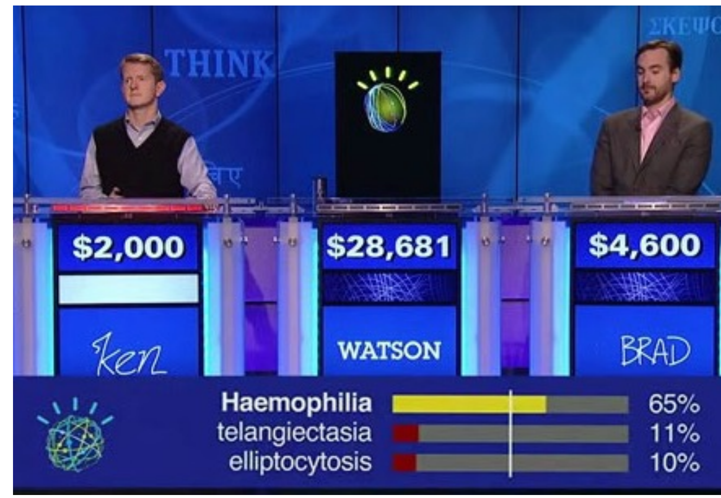
https://www.nytimes.com/2014/11/18/science/researchers-announce-breakthrough-in-content-recognition-software.html

Question → Answers







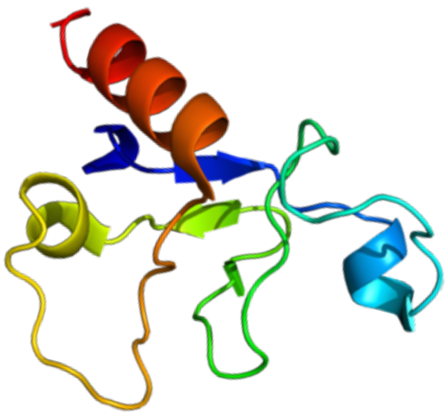


integrate with knowledge systems and natural language

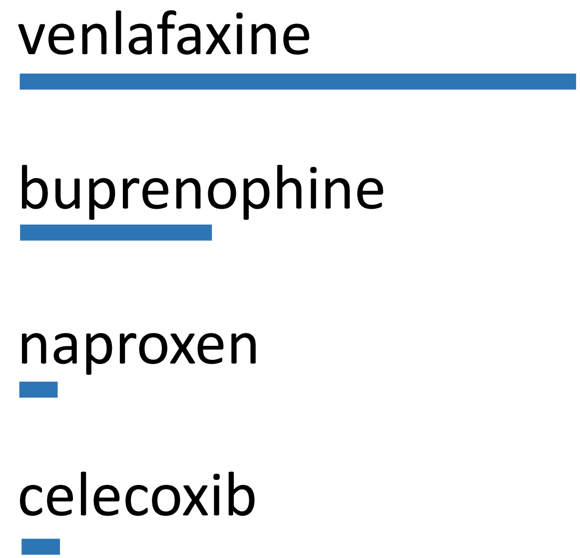
IBM Watson on Jeopardy game show:
https://www.youtube.com/watch?v=P18EdAKuC1U

Molecule → Activity

MERCK
Be well



?



Completed • $40,000 • 236 teams

Merck Molecular Activity Challenge

Thu 16 Aug 2012 – Tue 16 Oct 2012 (23 months ago)

Important goal: understand your tools!!

sklearn.svm.SVC

Super important tool!!

class `sklearn.svm.` **SVC** (*C=1.0, kernel='rbf', degree=3, gamma='auto_deprecated', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None*) [source]

? ? ?

Parameters:

? **C** : ***float, optional (default=1.0)***

Penalty parameter C of the error term.

You cannot use software tools like this if you do not know how to configure them to your problem!

? **kernel** : ***string, optional (default='rbf')***

Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable.

? ? ?

? **degree** : ***int, optional (default=3)***

Degree of the polynomial kernel function ('poly'). Ignored by all other kernels.

? **gamma** : ***float, optional (default='auto')***

Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

? **coef0** : ***float, optional (default=0.0)***

Independent term in kernel function. It is only significant in 'poly' and 'sigmoid'.

Better intro in Moodle!

Designed to help software engineers understand ML and what some of the big ideas are about.

Introduction to Machine Learning

Machine Learning
Concordia University

What is machine learning?

Machine learning can be understood from several perspectives. This course emphasizes a software perspective: that *machine learning is just an example-driven way to build programs.* Machine learning is an important approach to software development because it can build programs that no human could write by hand, and because those programs are *valuable*. Programs built by machine learning can do new things, earn more money, save more time, and even save more lives than programs built by traditional software engineering methods.

Machine learning is also hard to do correctly, and it is easy to fool yourself into thinking that you have succeeded in building a useful program—or a safe program—when in fact you have not. That is why software engineers should formally study machine learning!

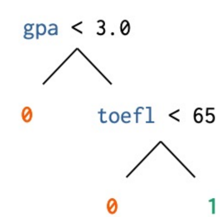
What is machine learning to a software engineer?

Humans are amazingly good at learning ideas from just a few examples. So, the best way for you to understand the idea of machine learning will be *by example*! Let's start with a simple example and then build on it.

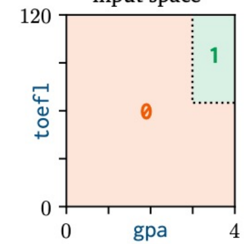
To see what it means to "build a program" in an "example-driven" way, first try to imagine writing the example program below the usual way, by hand. The program either accepts (`1`) or rejects (`0`) a student from university based on grades (GPA) and language proficiency score (TOEFL). At left is the C code, and at right are two depictions of how it works: as a *decision tree* diagram, and as a plot of outputs (`0` or `1`) over the input space.

```
int accept(float gpa, int toefl) {
    if (gpa < 3.0)
        return 0;          // REJECT
    else if (toefl < 65)
        return 0;          // REJECT
    else
        return 1;          // ACCEPT
}
```

decision tree



input space



As a human, to create the above program you would have to: understand the goals and requirements; design steps that hopefully achieve those goals; express those steps in a human-readable programming language. That is how software is made, the usual way.