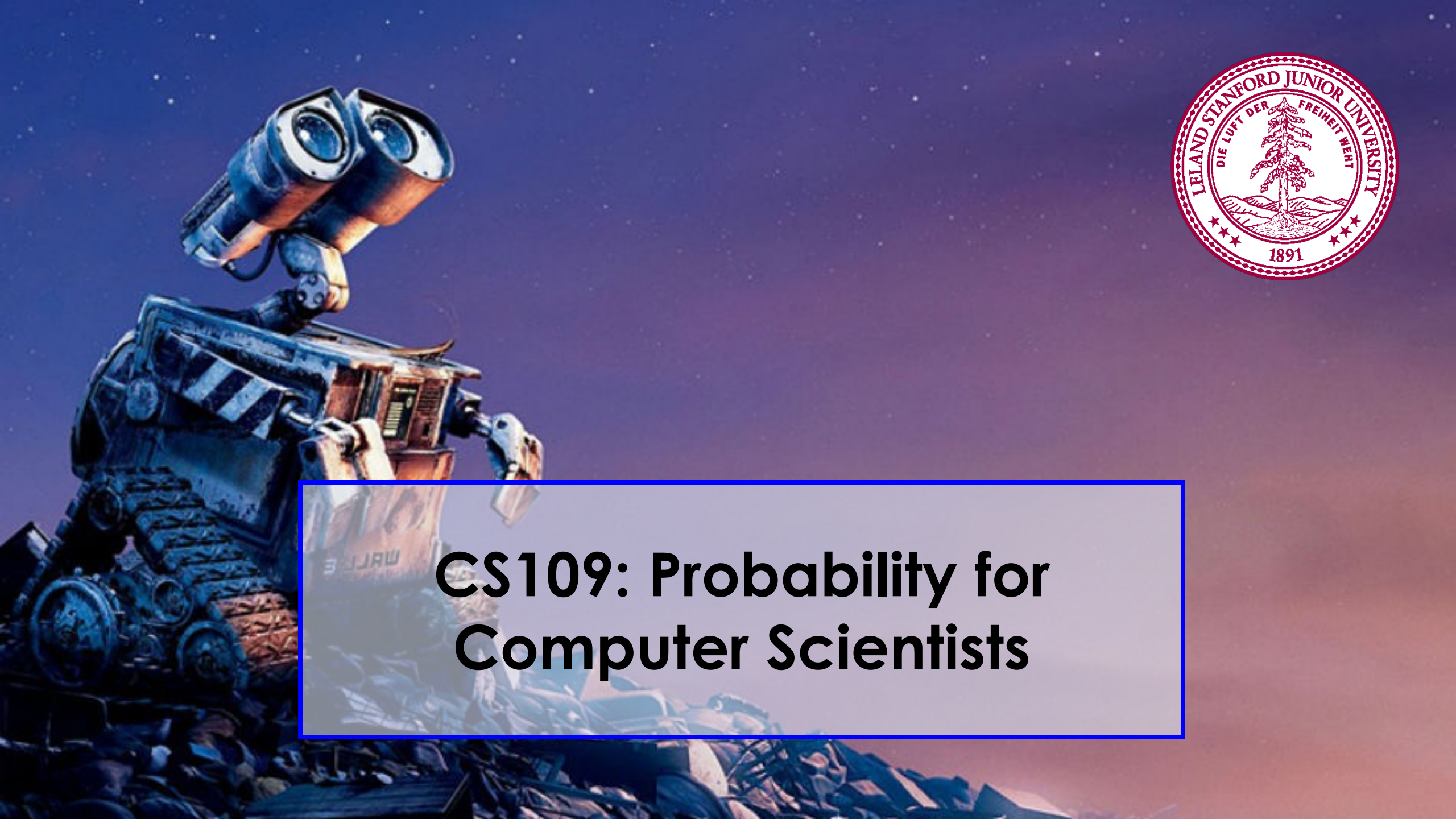


CS109: Probability for Computer Scientists


Probabilistic Models
Uncertainty Theory
Machine Learning
Random Variables
Core Probability
Counting
The Journey of CS109:

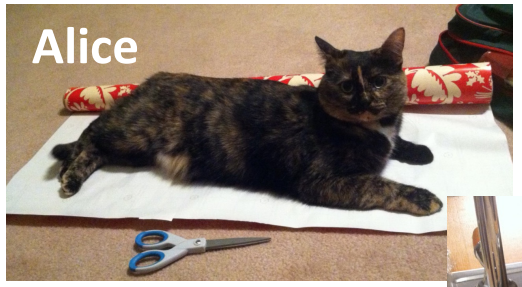
But first...

Hi! I am Kelly ☺

Hi! I am Kelly ☺

I grew up in Tampa, Florida

Hi! I am Kelly ☺

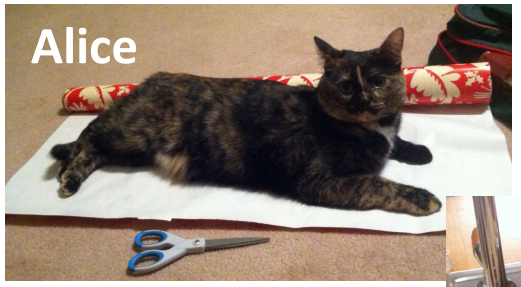


I grew up in Tampa, Florida

My family has 2 cats

Hi! I am Kelly ☺

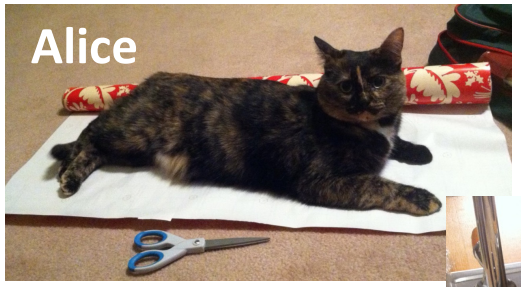
I grew up in Tampa, Florida



My family has 2 cats

I have always been a music nerd

Hi! I am Kelly ☺



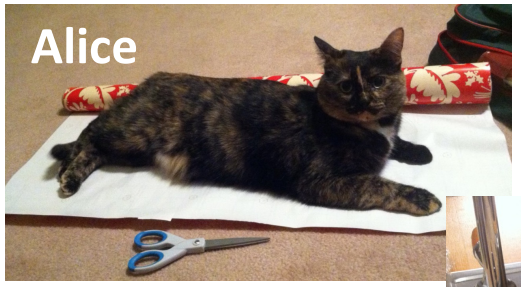
I grew up in Tampa, Florida

My family has 2 cats

I have always been a music nerd

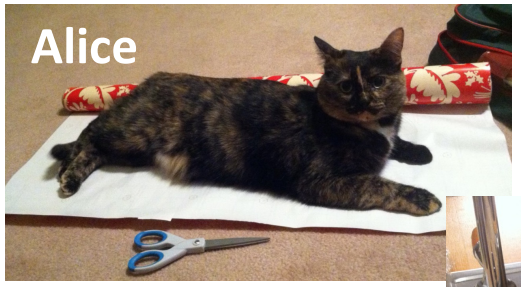
…and maybe just a nerd in general

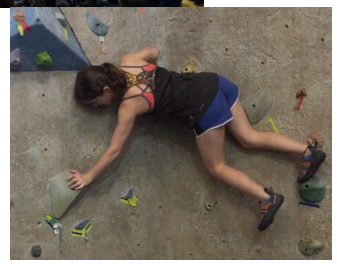
Hi! I am Kelly ☺

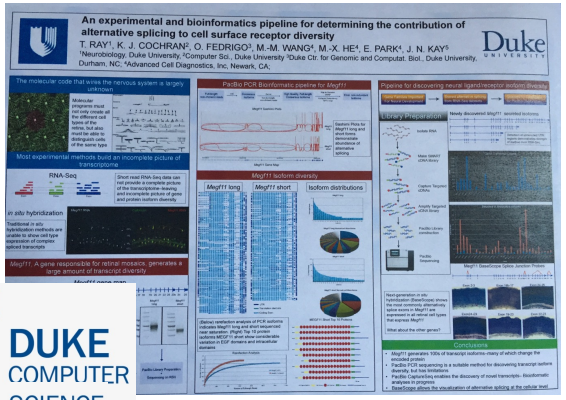


I went to Duke for undergrad

Hi! I am Kelly ☺

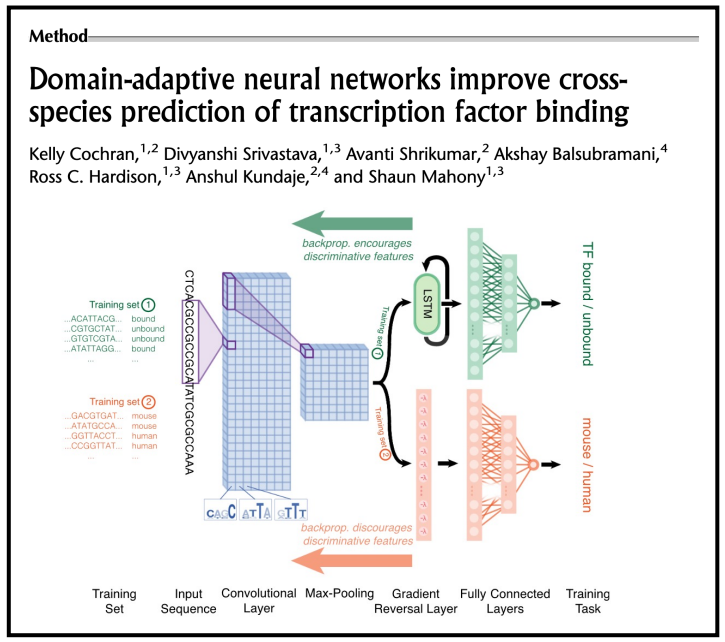


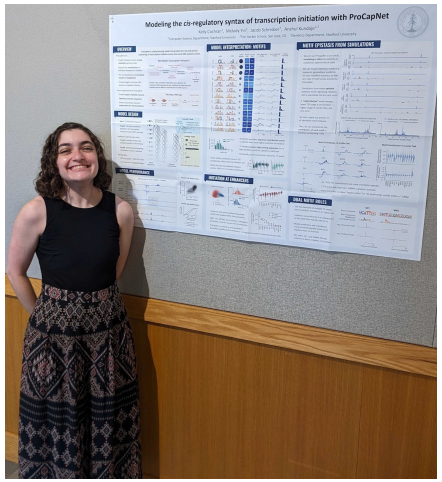




I went to Duke for undergrad – lots of music, basketball, rock climbing, learning, and research

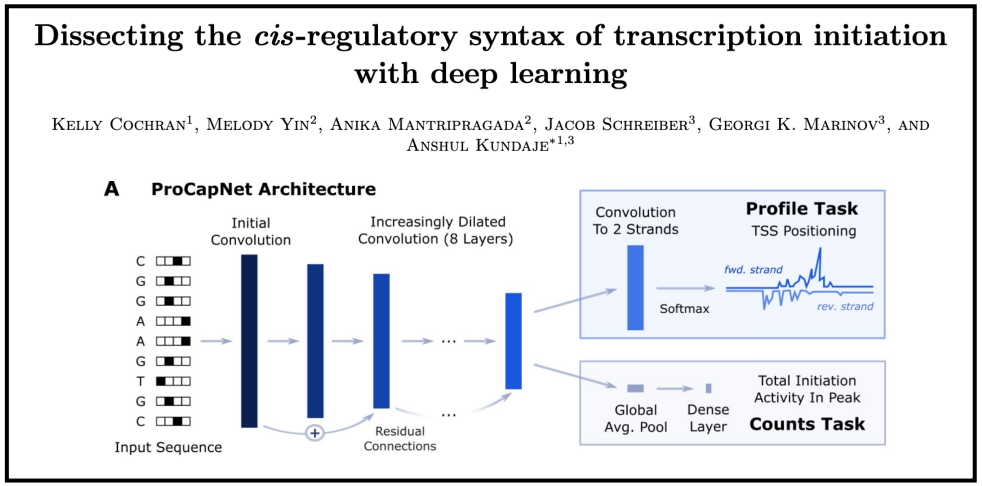
My Time At Stanford



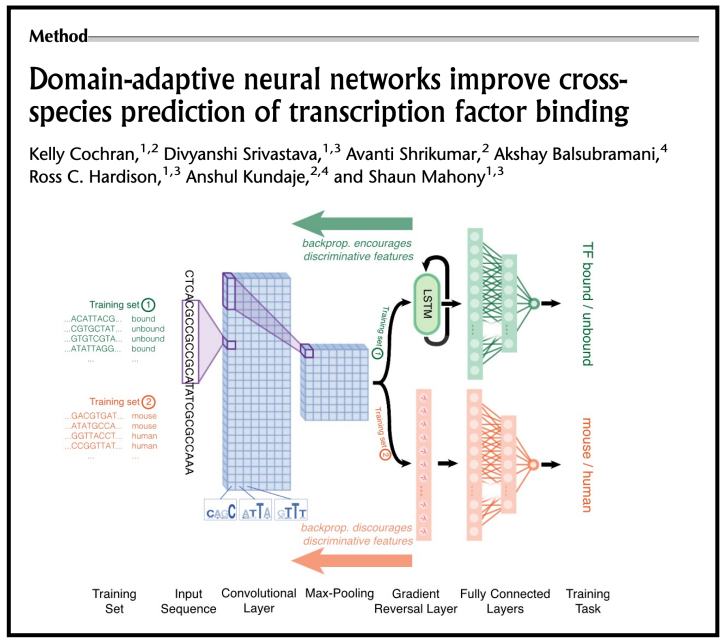


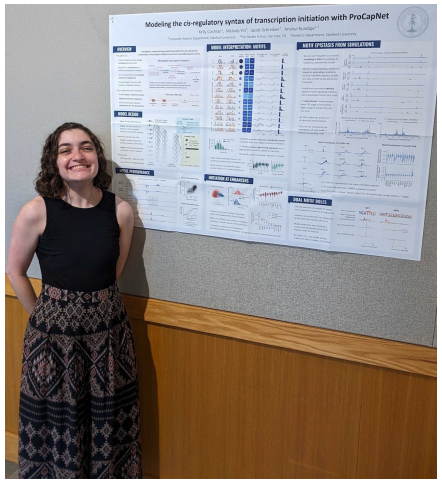
Mostly Research!

My PhD is in deep learning for genomics (biology)

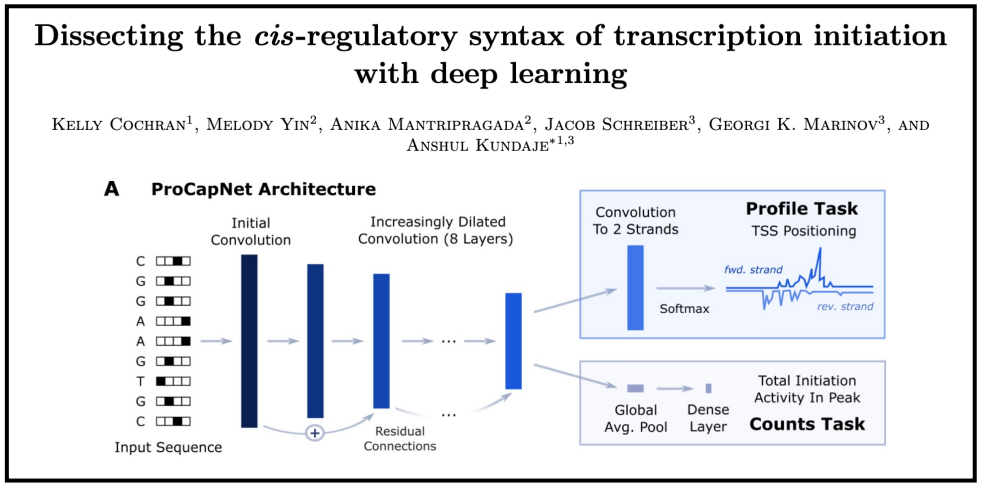


My Time At Stanford

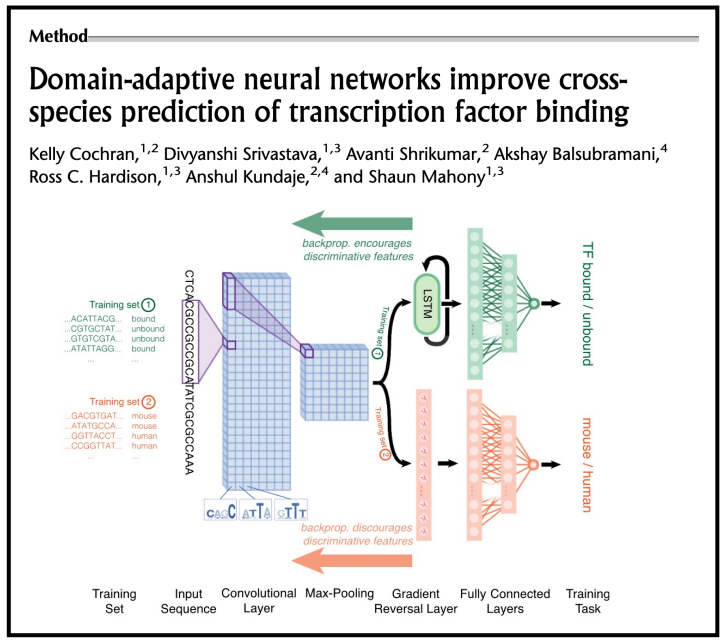


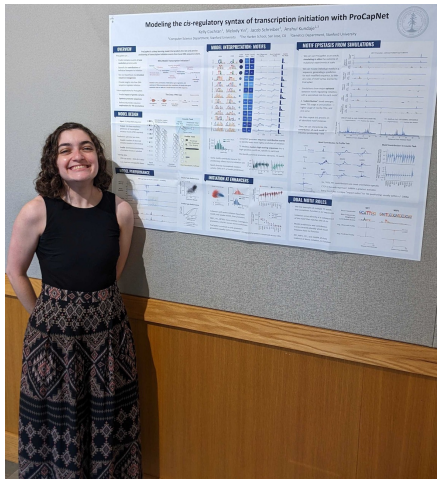


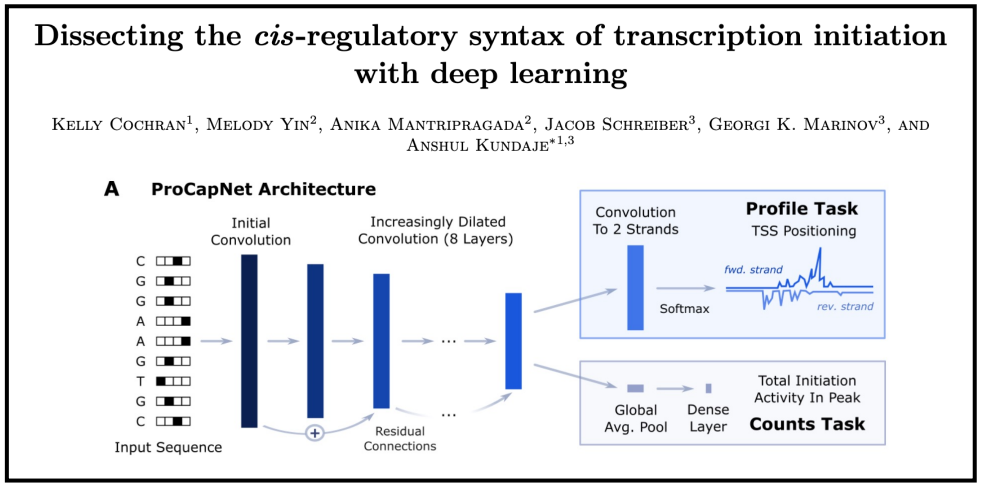
Also Teaching!
(CS109 is my favorite)

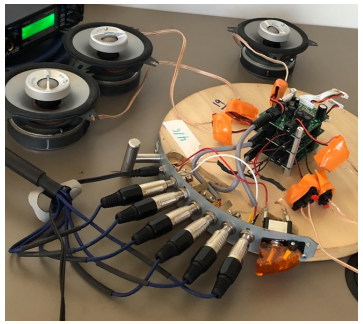


My Time At Stanford









Stanford Laptop Orchestra?!

Fantastic Teaching Team

Kelly Cochran
kcochran @ stanford dot edu
Co-Instructor

Joel Ramirez
joelramirez @ alumni dot stanford dot edu
Co-Instructor

Isabel Michel
imichel @ stanford dot edu
TA

Joel Ramirez

“Machine learning
on my mind”

Bachelors 2021 Symbolic Systems
Masters 2023 Computer Science

Where am I from?

Fort Worth, Texas

Hobbies?

Mariachi (Violin)

Where do I live?

San Francisco

Do I have Pets?

Miso Soup

Ube Donut

Joel Ramirez

e-mail: joel101@stanford.edu
phone: 682-331-3934
office: durand 319

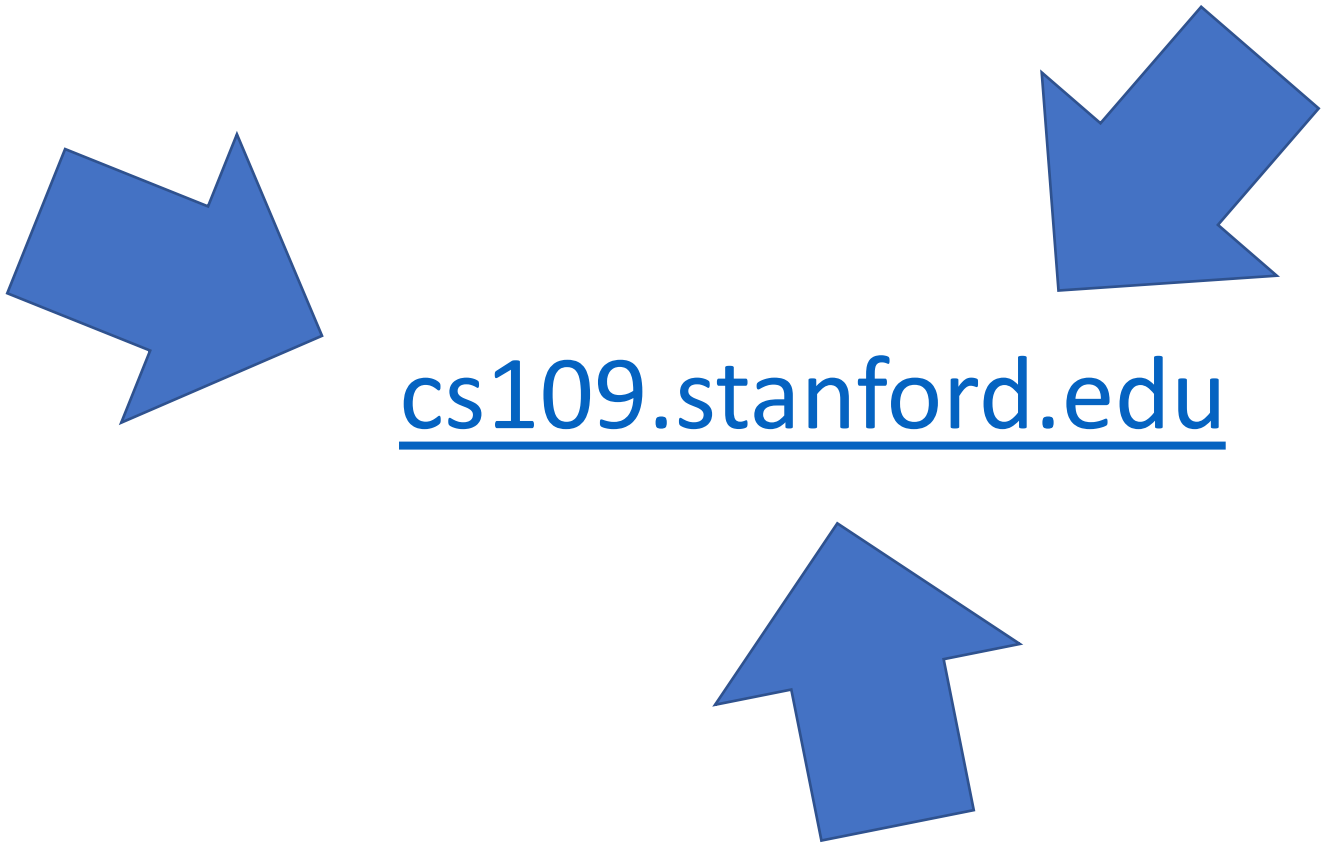
Super excited to work with y'all this quarter!!

Isabel’s Dog!!!

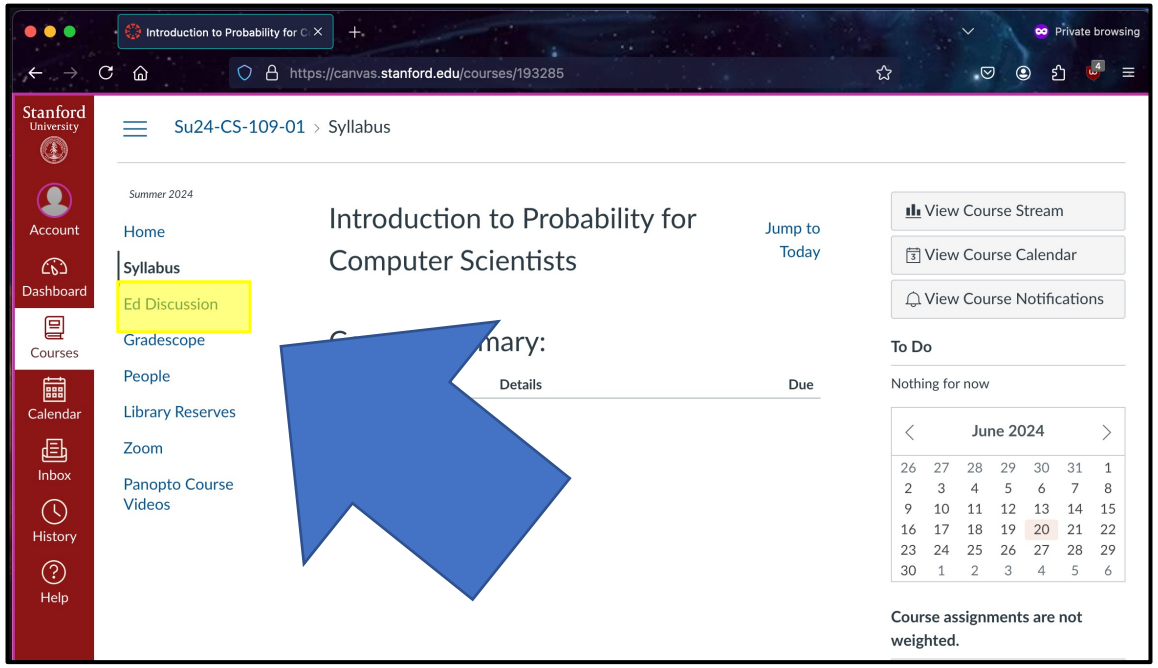
Course mechanics

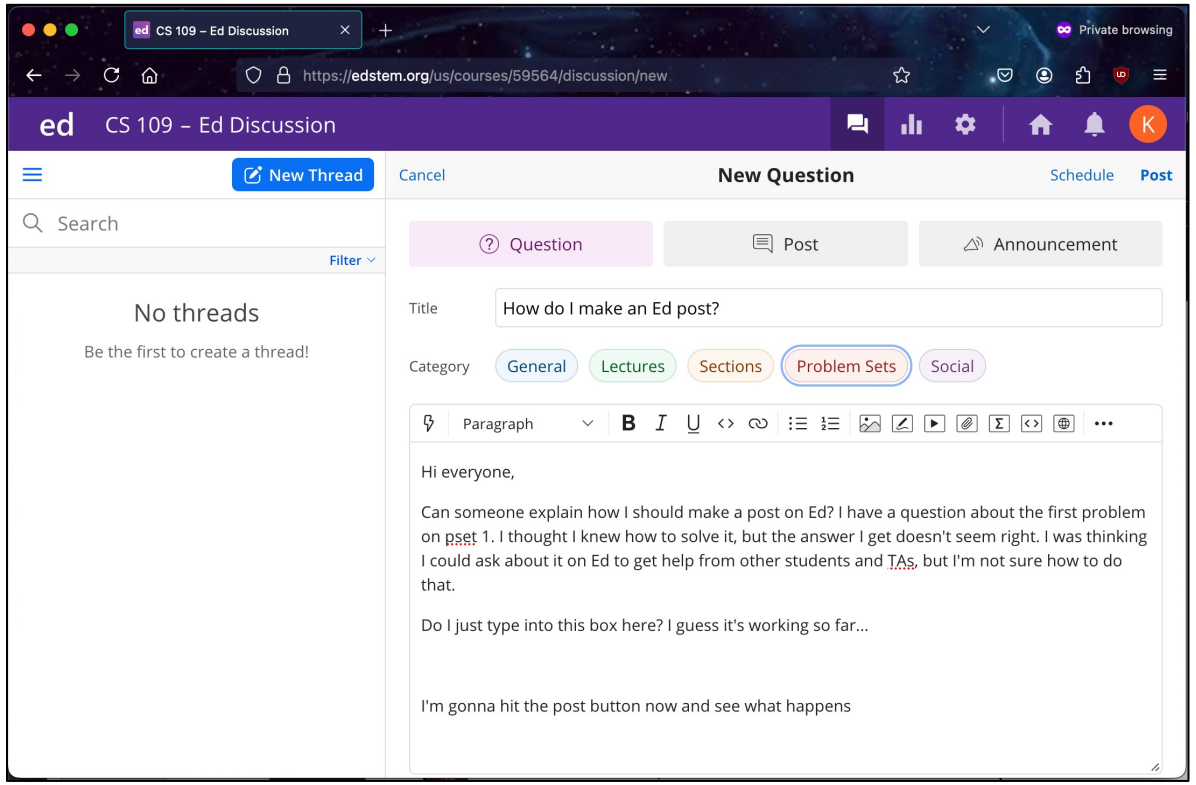
(this is a light version. Please read the syllabus for details.)

Most Info Is On The Course Website



Ed: Announcements, Ask Questions, Help Others

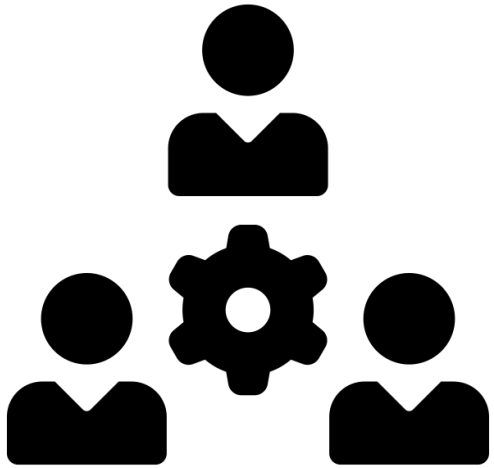




- Ed is the best place outside office hours to get unstuck on problems or ask any other questions
- Keep questions “public” when possible (so other students can answer)
- …but don’t copy-paste code or give away answers in public posts!

How To Get Help

Q&A forum
All announcements

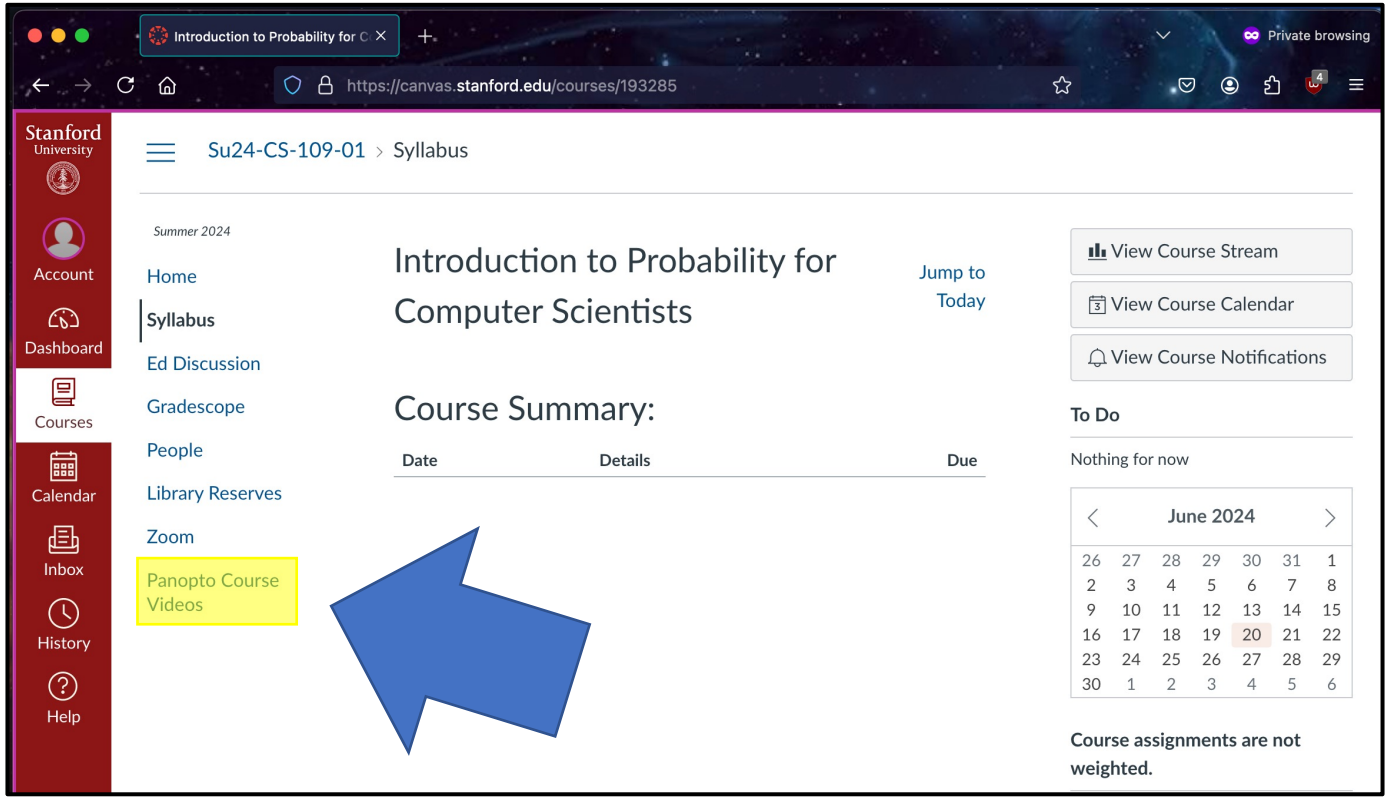


“Working” office hours
start on Saturday

cs109@cs.stanford.edu

Kelly and Joel have extra 1-on-1 office hours

Are Lectures Recorded? Yes



BUT *you should come to class* – you'll get extra credit for attendance!

Coming to class → not falling behind on lectures → finishing homework on time → kicking butt on exams → 😎

Class Breakdown

40% **5 Problem Sets**

20% **Midterm**
2-hour exam: Tuesday, July 23rd, 7pm

30% **Final**
3-hour exam, Saturday, August 17th, 3:30pm

10% **Section Participation** Sign up by Wednesday!

+3% **Lecture Attendance** (extra credit)

We Use The Psetapp!

Pset 1 - Counting for Probability

Private browsing

https://cs109psets.netlify.app/sum24/pset1/dnaturns

PS1

1 2a 2b 2c 3 4 5a 5b 6 7 8

DNA Turns

A DNA-turn has 10 base pairs. Each base pair can take on one of four distinct values, (A, T, G, C). How many distinct DNA-turns of length 10 are there?

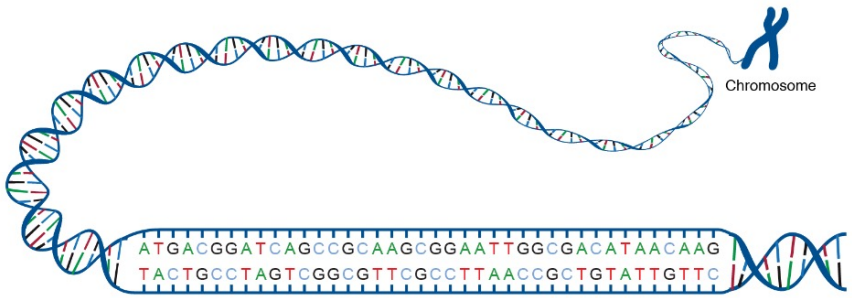


Figure: An example DNA strand, this one is length 40 (read the top row of letters, the bottom is a matching base pair)

Answer Editor | Solution

Numeric Answer: Enter your answer | Check Answer

Explanation:

Block LaTeX | Inline LaTeX | Python | Image

Here's how I calculated my answer! I used this rule. I took the number 10 and the number 4 and...

```
\begin{aligned}
    1 + 2 * 3 ^ 4 = 5
\end{aligned}
```

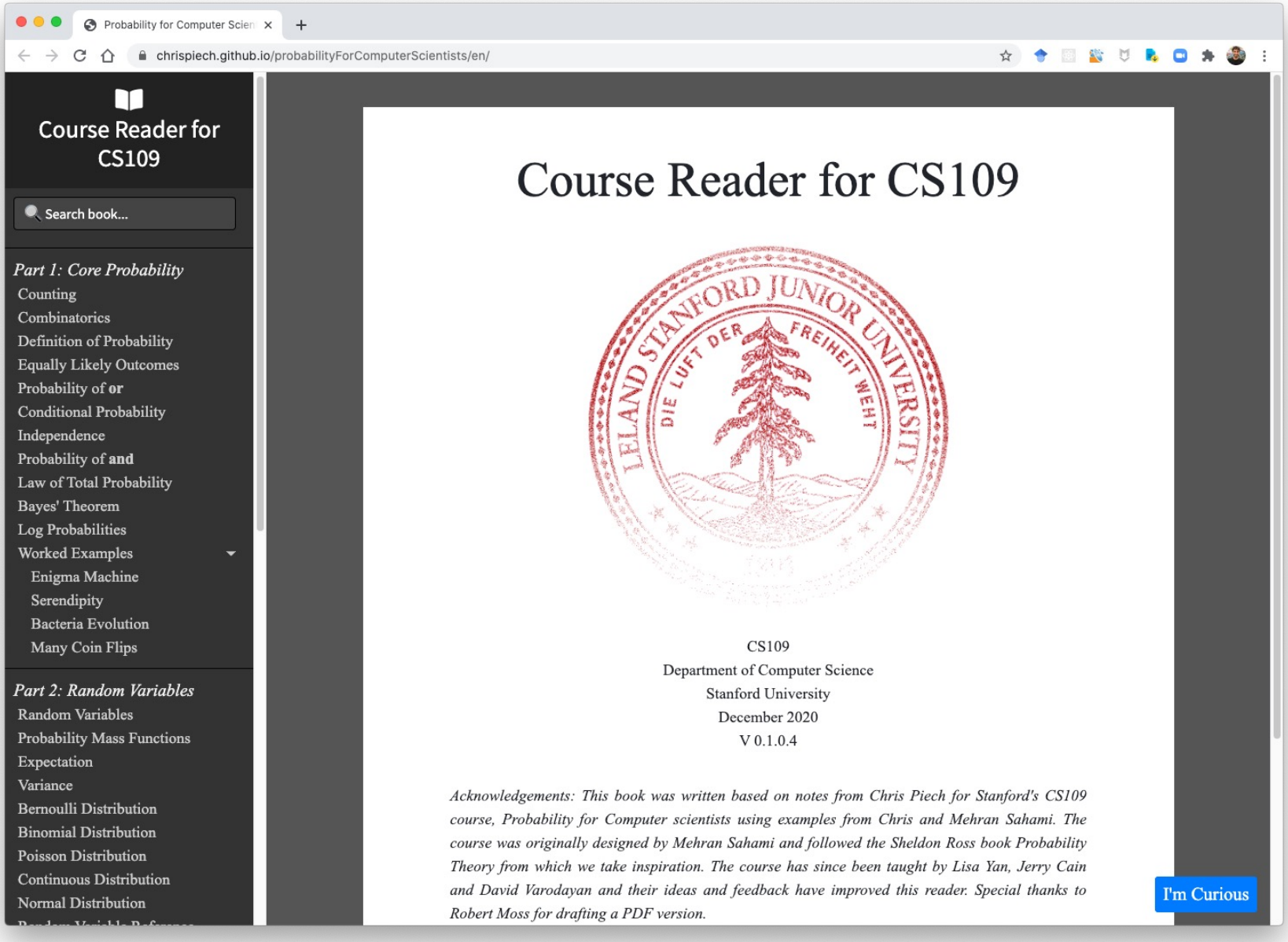
$$1 + 2 * 3^4 = 5$$

Close Block Editor

```
# your code here!
print(1 + 2 * 3 ** 4)
```

Run | Show

We Have a (Super Cool) Course Reader!



Prerequisites

What do you really need?

CS106B/X (important):
- Hash Tables
- Recursion
- Binary Trees
- General programming chops

CS103 (not necessary):
- Set theory
- Math maturity
- Proof techniques, induction (no hard proofs in 109)

Math 51 or CME 100 (important, co-req ok)
- Derivatives, multivariate differentiation
- Integrals, multivariate integration
- Basic familiarity with linear algebra basics (vectors)

Assignments include writing Python code

Review session this week!

How Many Units Should You Enroll In?

- If you are a **Stanford undergraduate**: 5 units
- If you are a **Stanford grad student**: you can choose 3, 4, or 5
  - There is no difference in workload or grading based on units
- If you are a **visiting student**: you should be good to go

Stanford's rule is units x 3 = # hours/week spent on class

- For CS109, this translates to ~10 hours/week on problem sets + exam prep

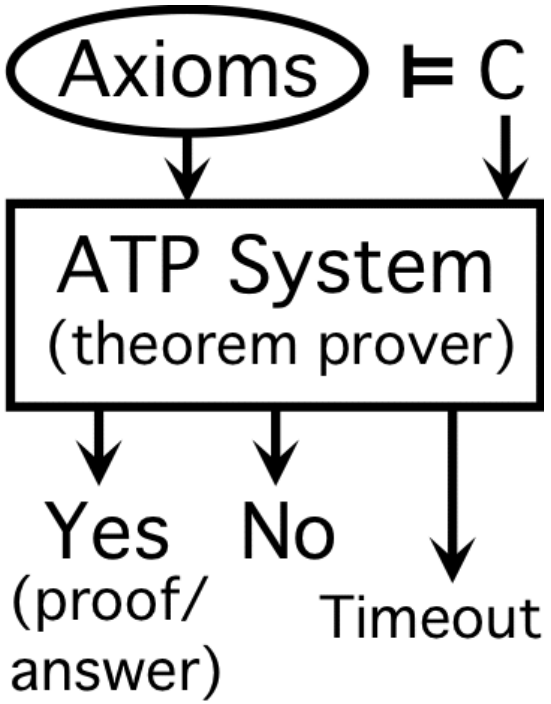
Let’s start with a story…

Modern AI
or, How we learned to combine probability and programming

Early AI Optimism in the 1950s

1952

1955

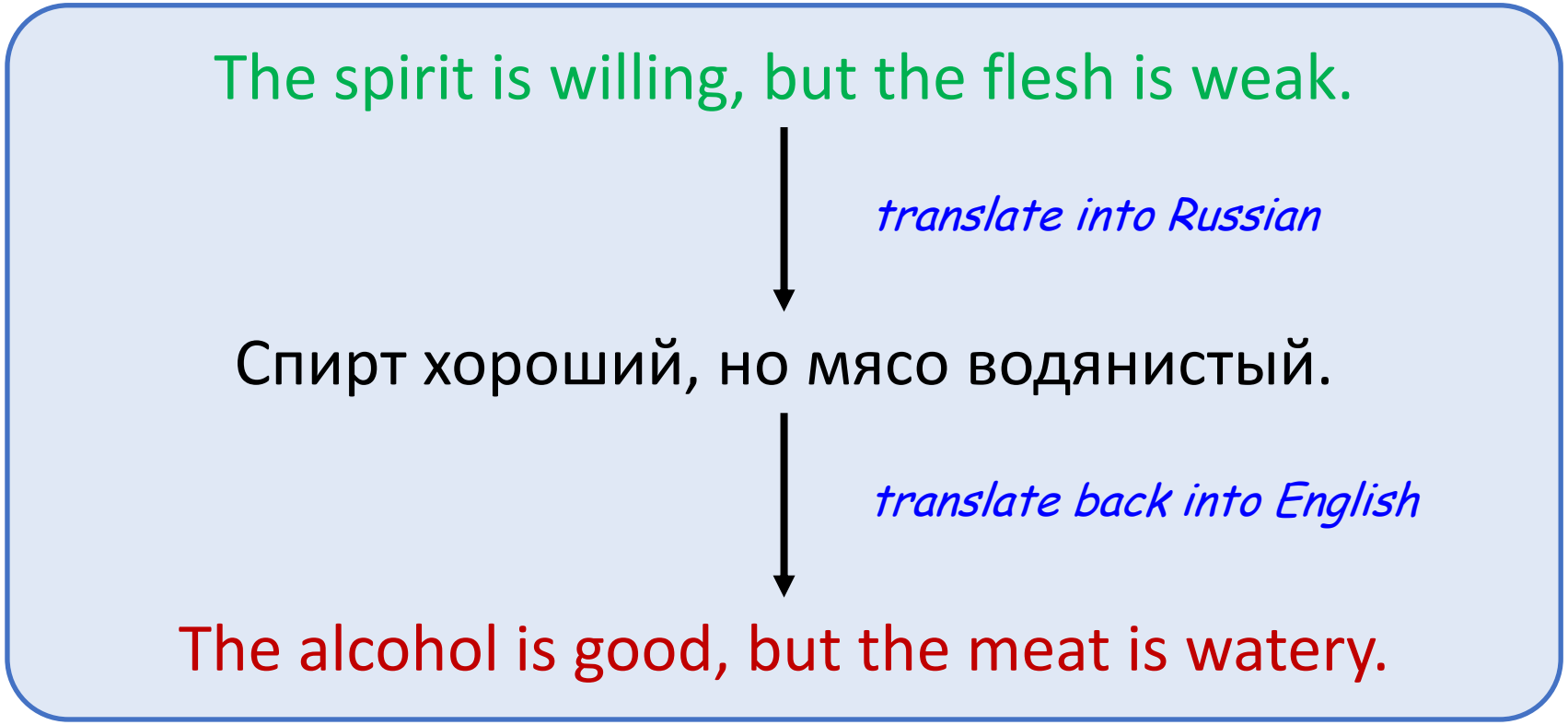


Early AI Optimism in the 1950s

"Machines will be capable, within twenty years, of doing any work a man can do."

–Herbert Simon, 1952

Underwhelming Results: 1950s to 1980s



Early AI research underestimated the complexity of the world

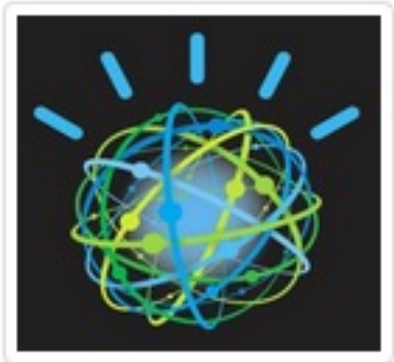

BRACE YOURSELVES
WINTER IS COMING

Fast forward…

Big Milestones Part 1

1997 Deep Blue

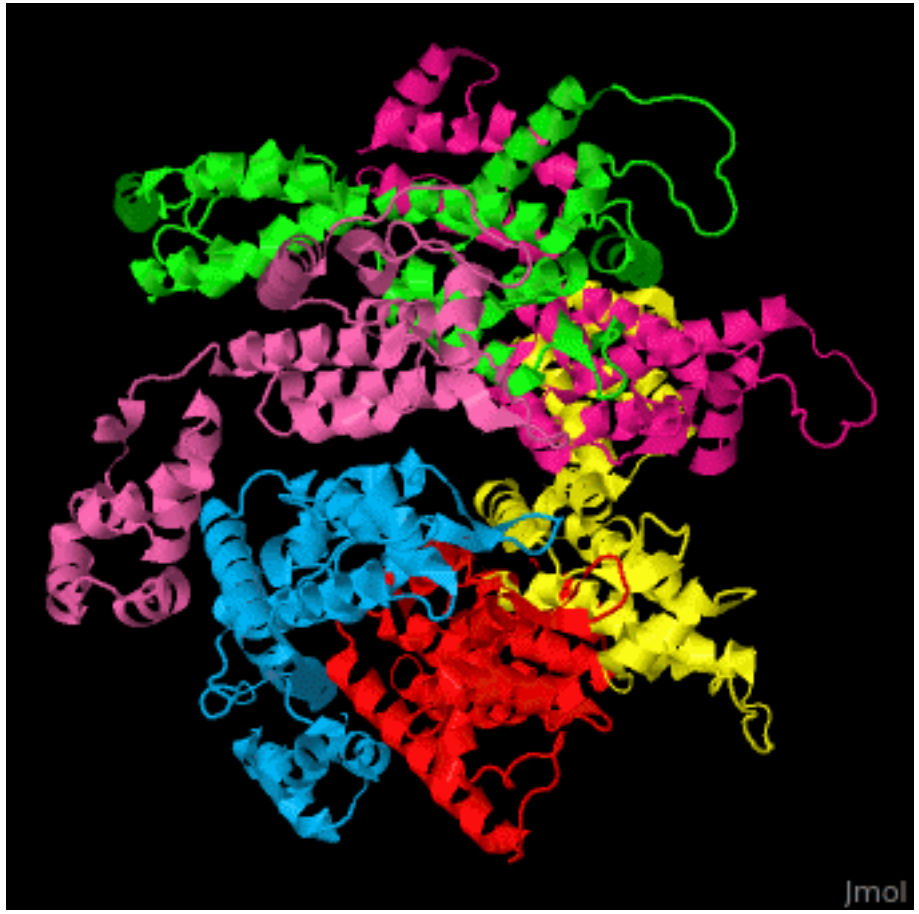
2005 Stanley

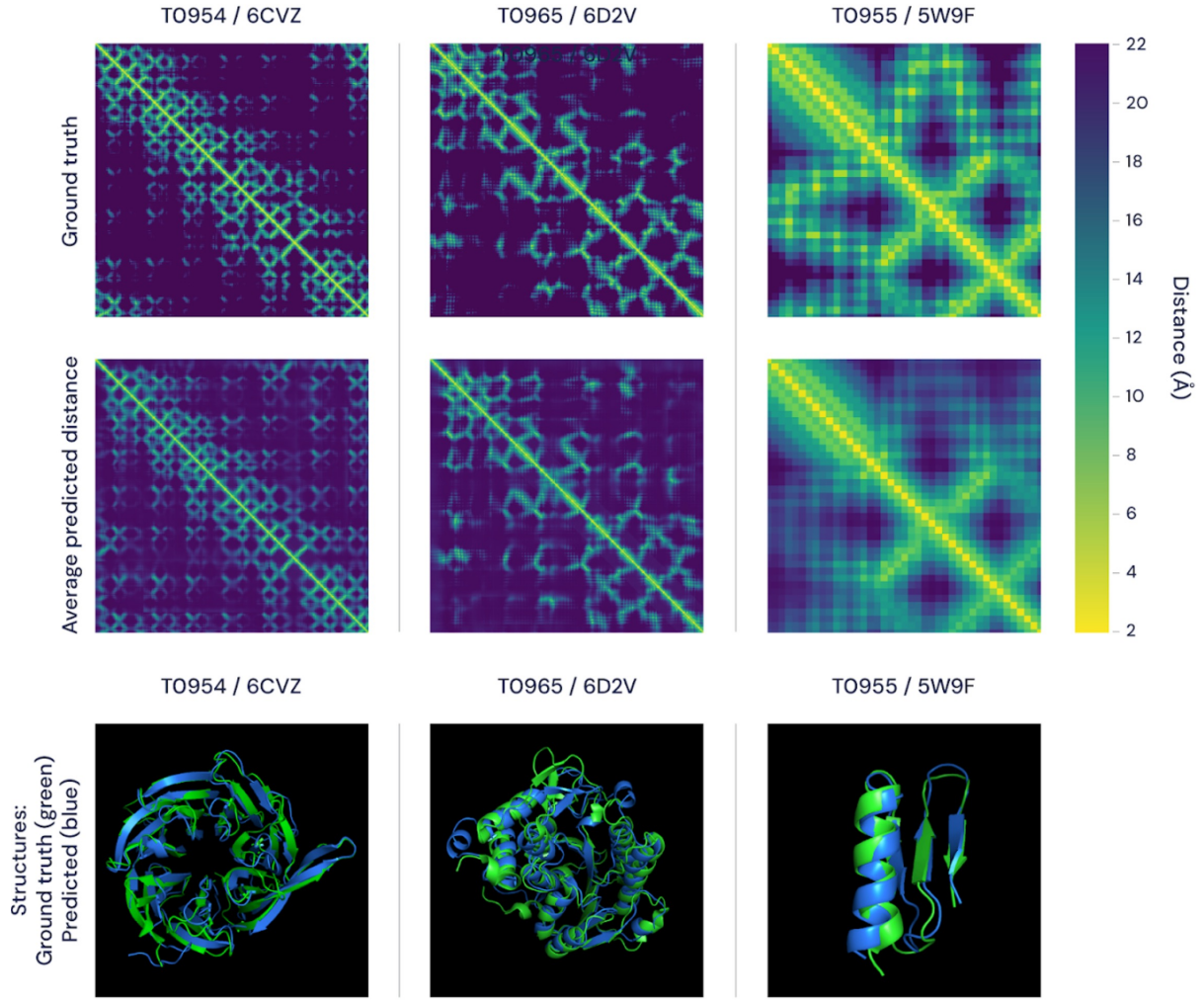


2011 Watson

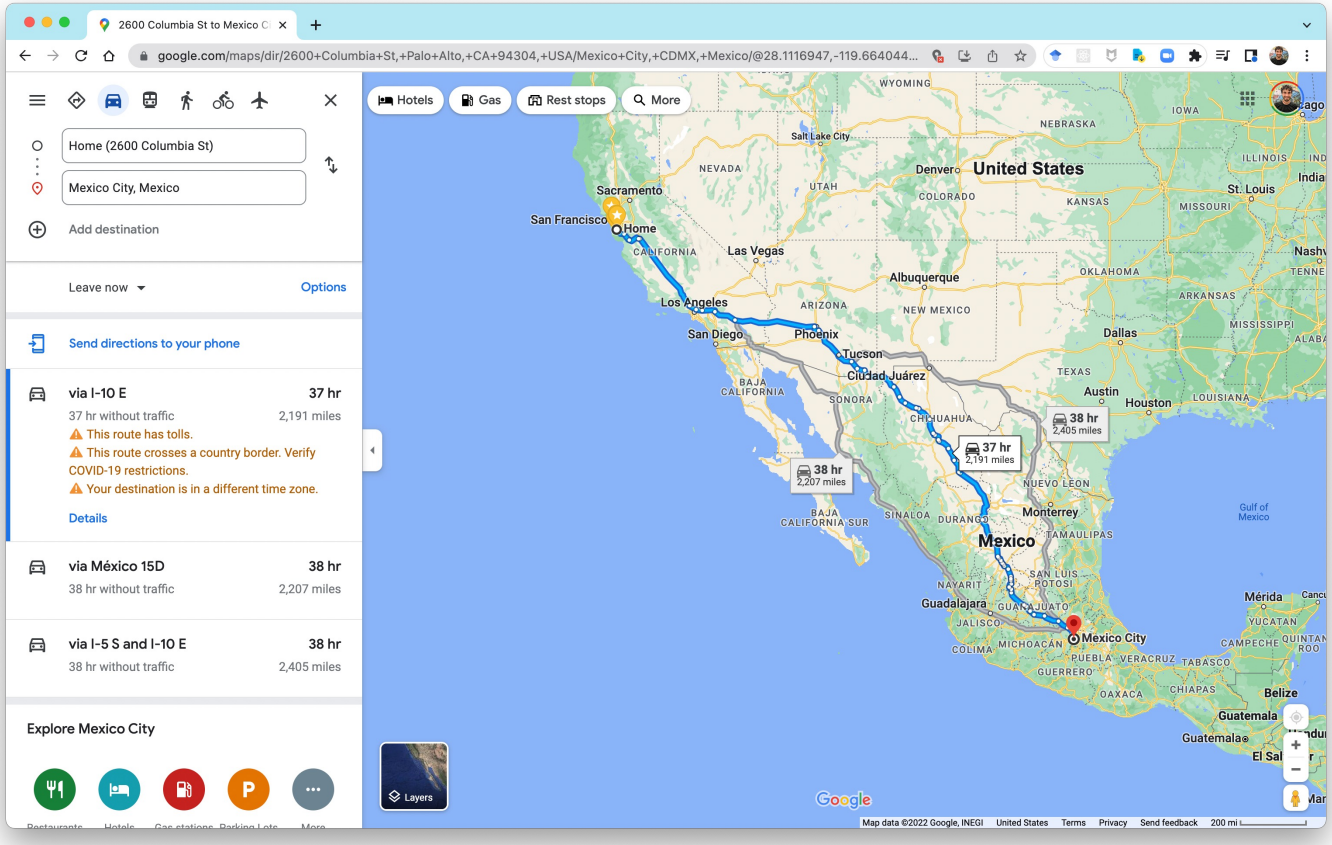
The last remaining board game: Go

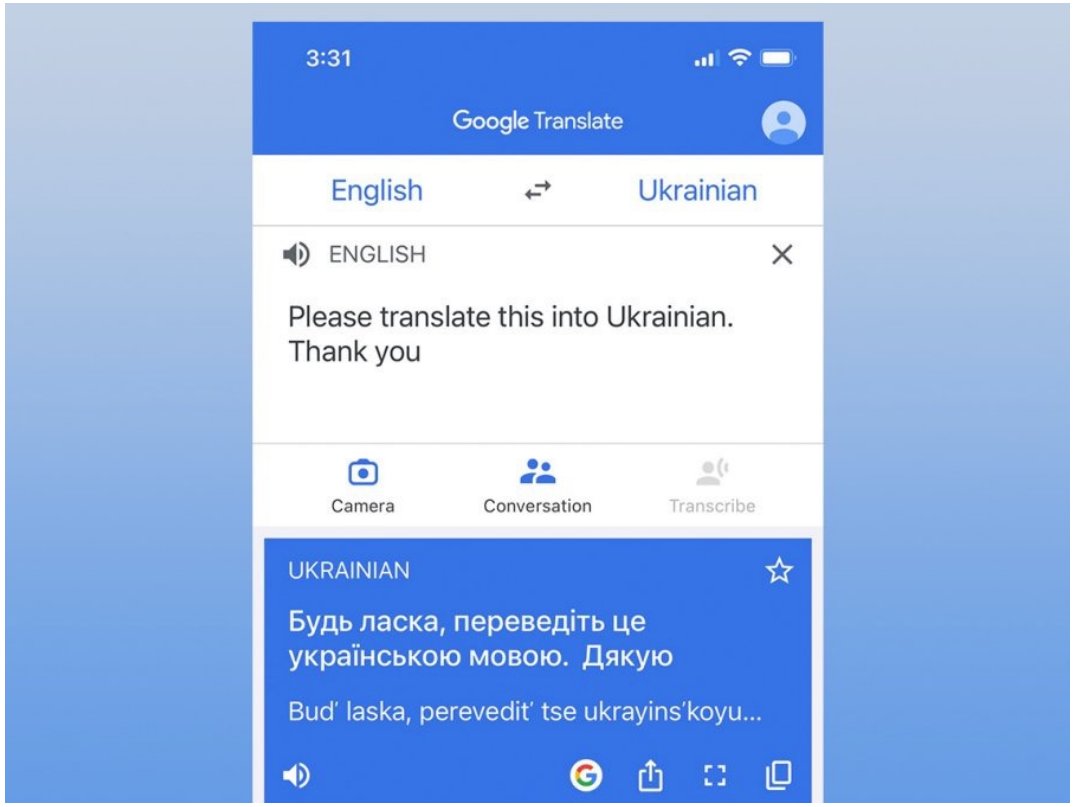
Scientific Advances: Protein Structure





Day-to-Day Algorithms





And then came ChatGPT

AI that can parrot human language

create the transscript of a programming meme

"Why do programmers always confuse Halloween and Christmas? Because Oct 31 equals Dec 25."

Transcript:

Programmer 1: Why do we always get Halloween and Christmas mixed up?
Programmer 2: I don't know, maybe it's because Oct 31 equals Dec 25.
Programmer 1: Oh, I see!
Programmer 2: Yeah, it's a classic programmer joke.
Programmer 1: Haha, I guess that makes sense.
Programmer 2: Yeah, it's just one of those things that we tend to remember.

i dont get it

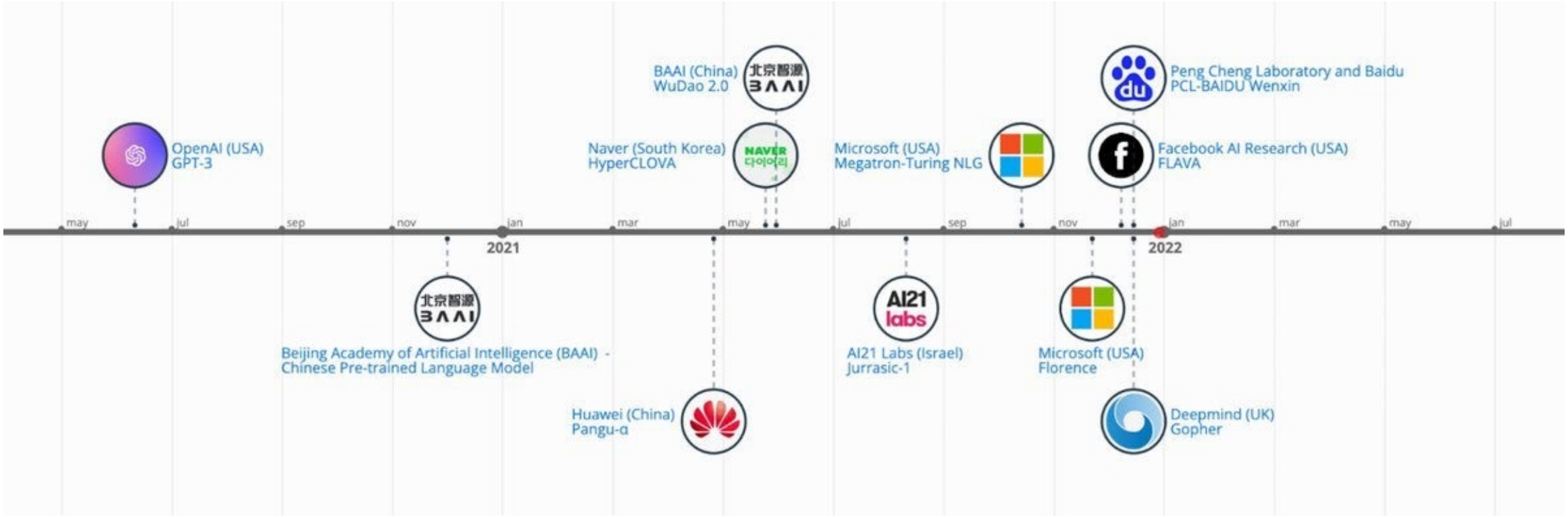
The joke is based on the fact that in many programming languages, the date October 31 is represented as 10/31, which can be shortened to 1031. Similarly, December 25 is represented as 12/25, which can be shortened to 1225. Therefore, 1031 and 1225 are easily confused, leading programmers to mix up Halloween and Christmas. It's a playful way of acknowledging the fact that programmers often have to work with numbers and dates, and sometimes get them confused.

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Authors: Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell | Authors Info & Claims

FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency
March 2021 • Pages 610 - 623 • https://doi.org/10.1145/3442188.3445922

Enter the LLM scaling race



AI-Generated Art

Dalle2. Prompt “a large lecture class at stanford learning probability for computer scientists in the style of vangough”

Genrative AI For Images

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However, these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function $\phi(\mathbf{x})$ yielding the flexible distribution $p(\mathbf{x}) = \frac{\phi(\mathbf{x})}{Z}$, where Z is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff–for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b;a), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvärinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective[1].

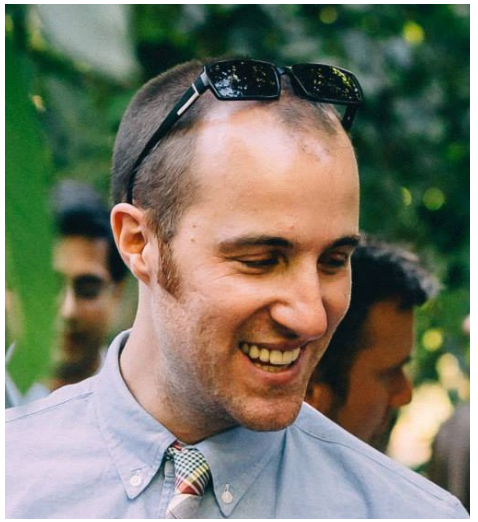
1.1. Diffusion probabilistic models

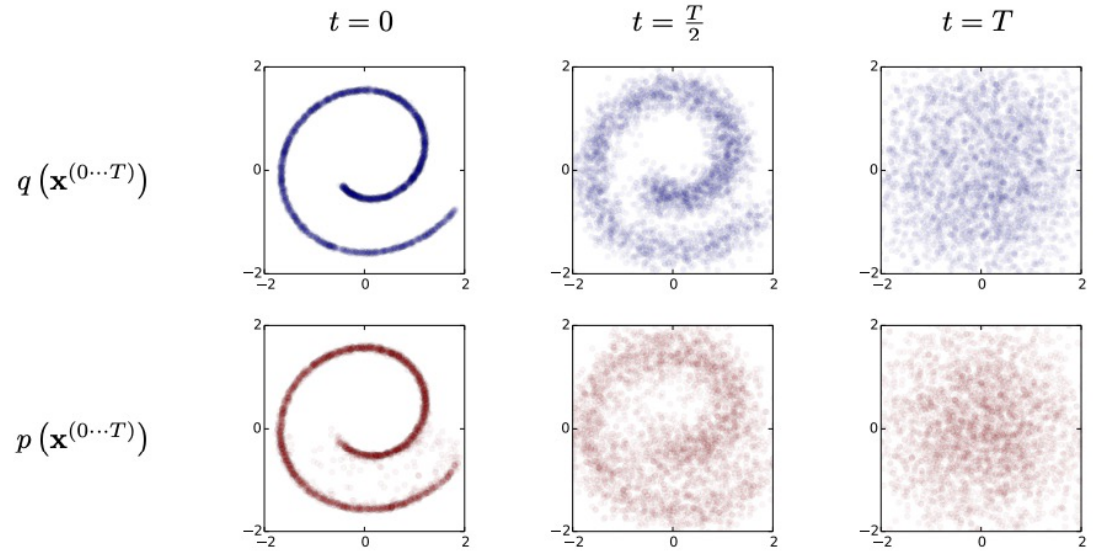
We present a novel way to define probabilistic models that allows:

1. extreme flexibility in model structure,
2. exact sampling,

[1]Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.





Piech & Cain, CS109, Stanford University

Where did all this tech come from?

Focus on one problem

Computer Vision

Chihuahua or muffin?

Can you do it?

Chihuahua or Muffin?

Chihuahua or Muffin?

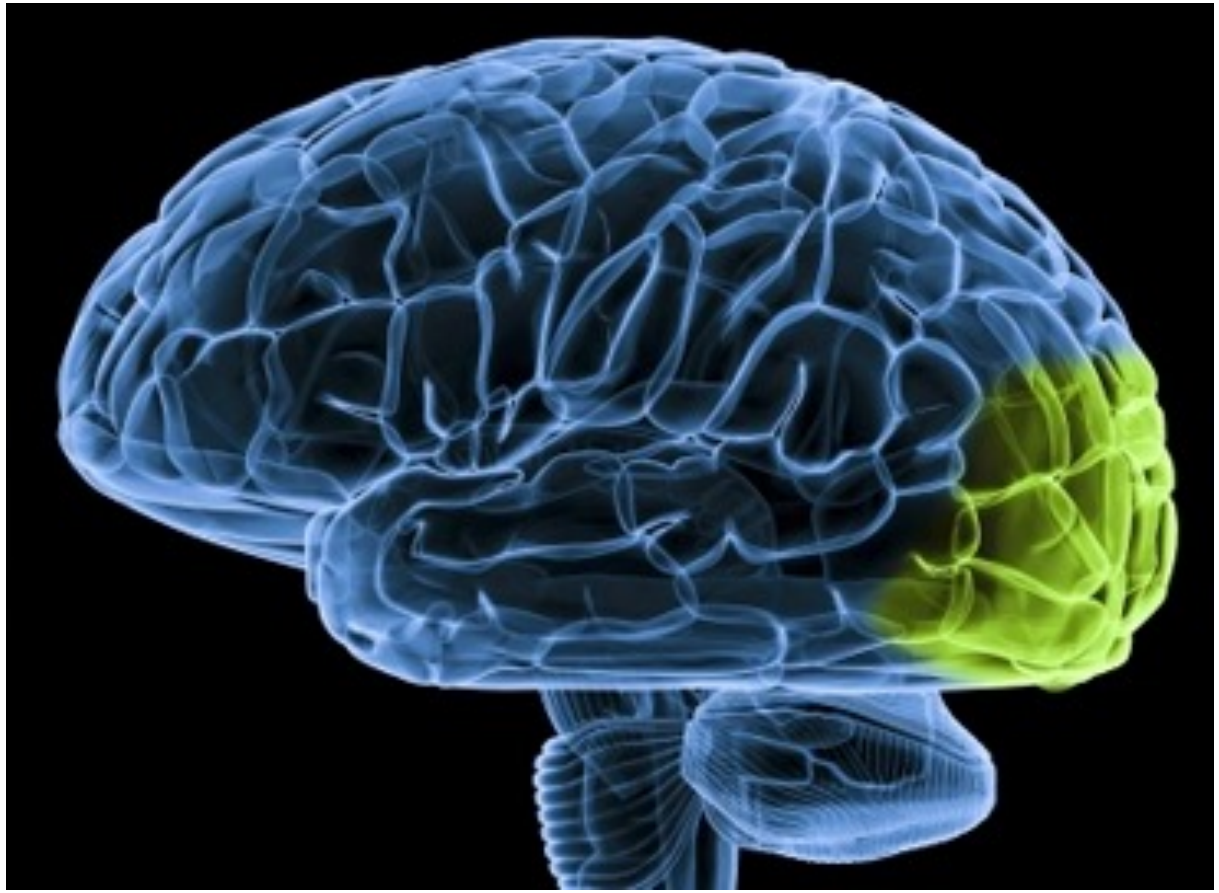
Chihuahua or Muffin?

What a computer sees

0	0	1	0	1	0	1	0	0	0	1	1	1	0	1
1	0	0	1	0	1	1	1	0	1	0	0	0	0	0
1	1	1	0	1	0	0	1	1	0	0	1	0	1	0
1	1	1	1	1	0	0	0	0	0	1	1	0	1	1
0	0	0	1	1	0	0	1	0	0	0	1	1	1	0
1	0	0	1	1	0	0	0	1	0	1	1	1	1	0
1	1	0	1	1	0	0	1	1	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	0	0	1	1	1	1
0	0	0	0	1	0	1	0	1	1	0	0	1	1	1
0	1	1	0	0	0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	1	0	0	0	1	0	0	0	0
0	1	1	1	0	1	0	0	1	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	1
0	0	1	1	1	0	1	0	1	1	0	0	0	1	0

What a human sees

Chihuahua or Muffin?



About 30% of your cortex is used for vision,
while only 3% is used to process hearing

This Reasoning Is Hard To Write As Code

```
def classify_chihuahua_or_muffin(list_of_pixels):
    # TODO...
```

0	0	1	0	1	0	1	0	0	0	1	1	1	0	1
1	0	0	1	0	1	1	1	0	1	0	0	0	0	0
1	1	1	0	1	0	0	1	1	0	0	1	0	1	0
1	1	1	1	1	0	0	0	0	0	1	1	0	1	1
0	0	0	1	1	0	0	1	0	0	0	1	1	1	0
1	0	0	1	1	0	0	0	1	0	1	1	1	1	0
1	1	0	1	1	0	0	1	1	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	0	0	1	1	1	1
0	0	0	0	1	0	1	0	1	1	0	0	1	1	1
0	1	1	0	0	0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	1	0	0	0	1	0	0	0	0
0	1	1	1	0	1	0	0	1	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	1
0	0	1	1	1	0	1	0	1	1	0	0	0	1	0

How does modern machine learning solve this?

Two Great Ideas

1. Artificial Neurons

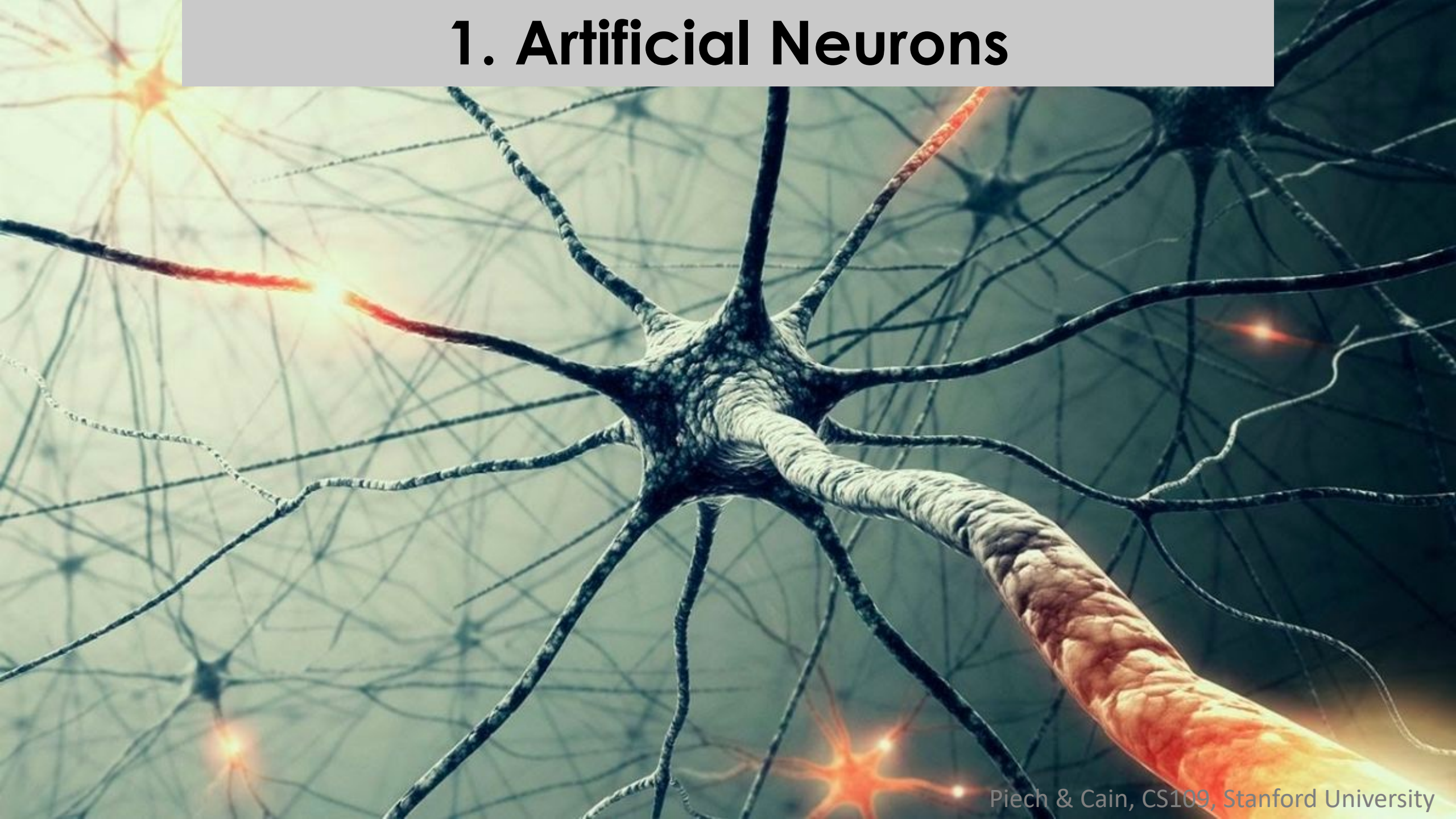
2. Learn by Example

Two Great Ideas

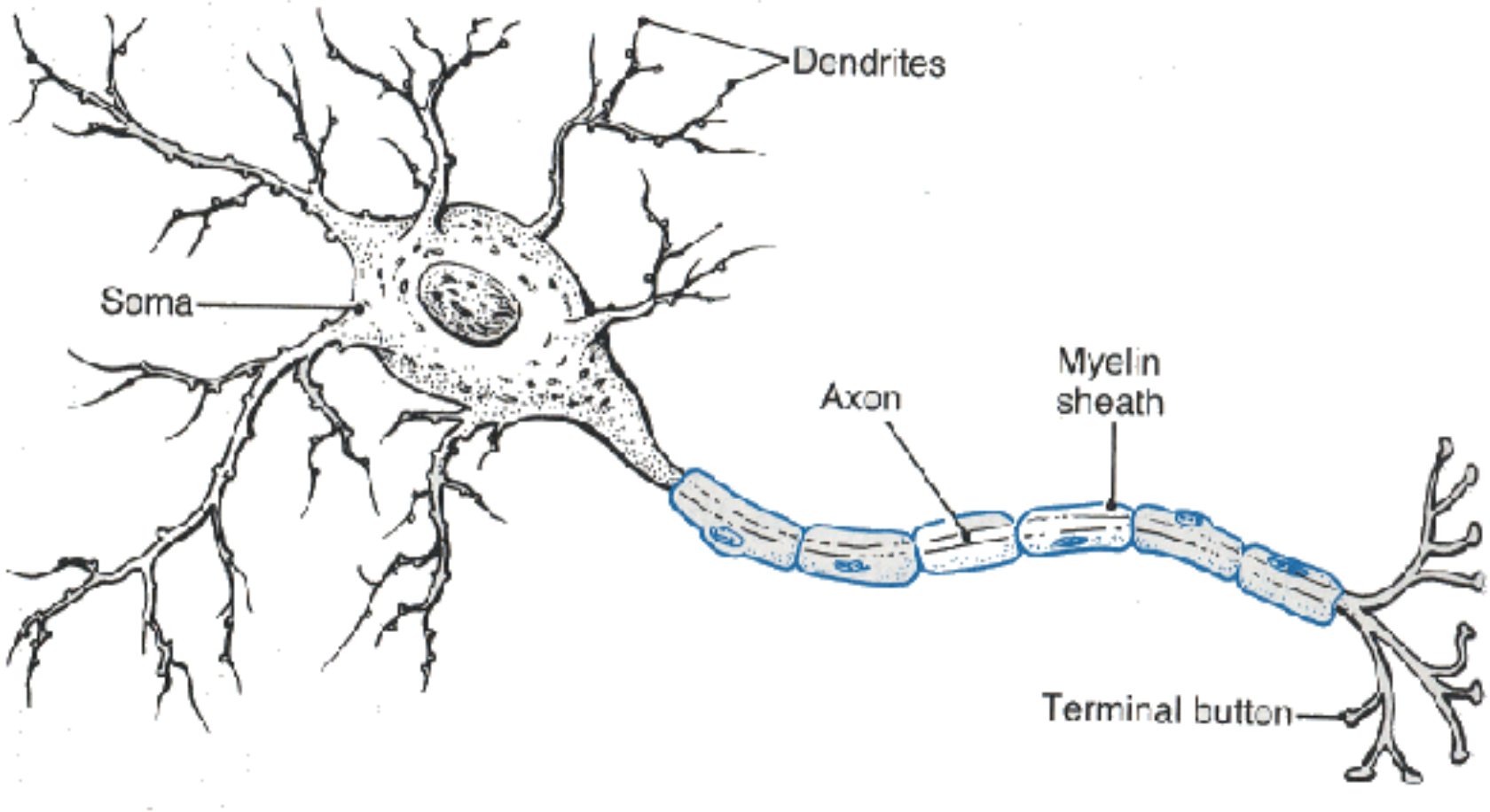
1. Artificial Neurons

2. Learn by Example

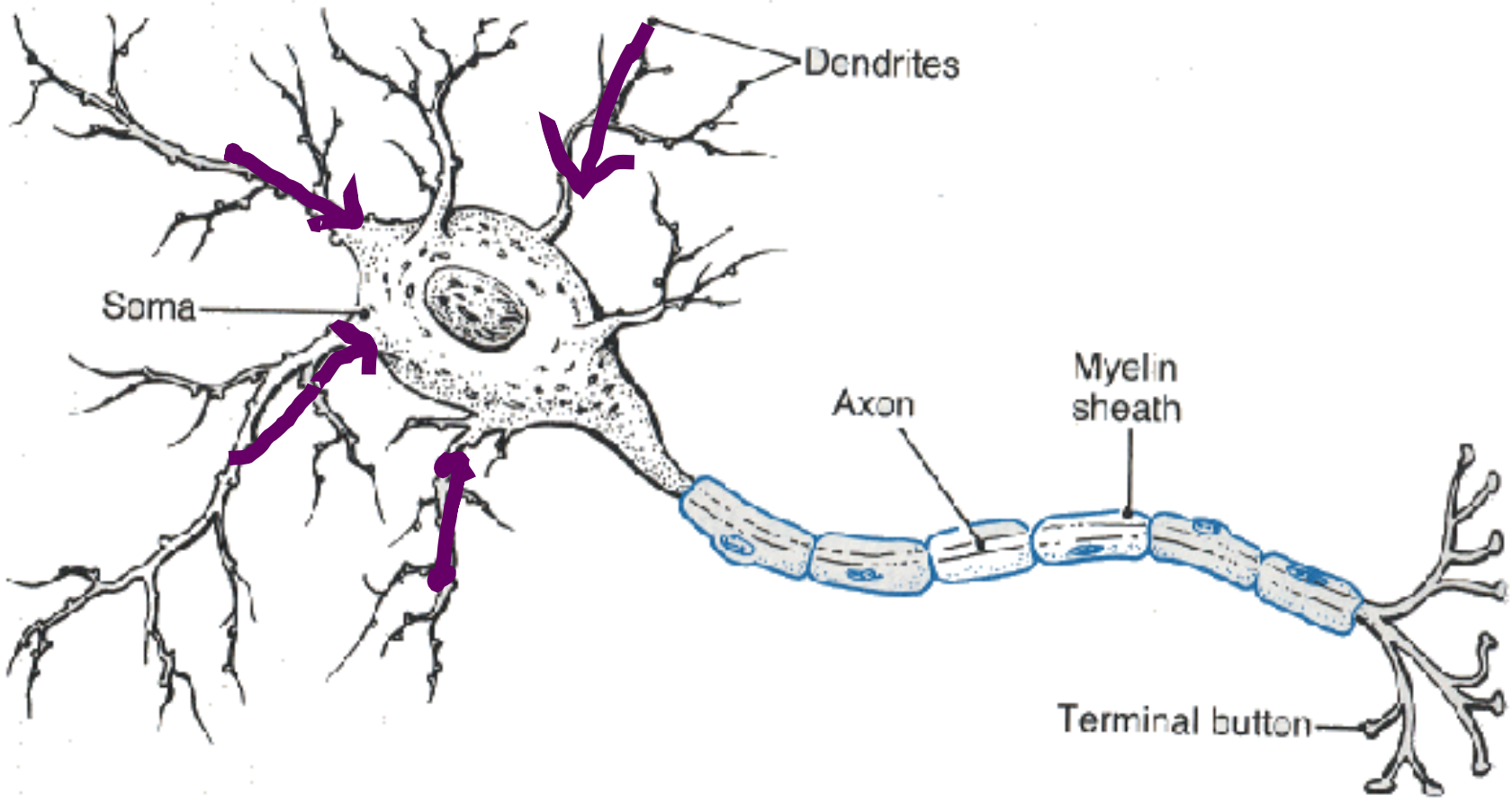
1. Artificial Neurons



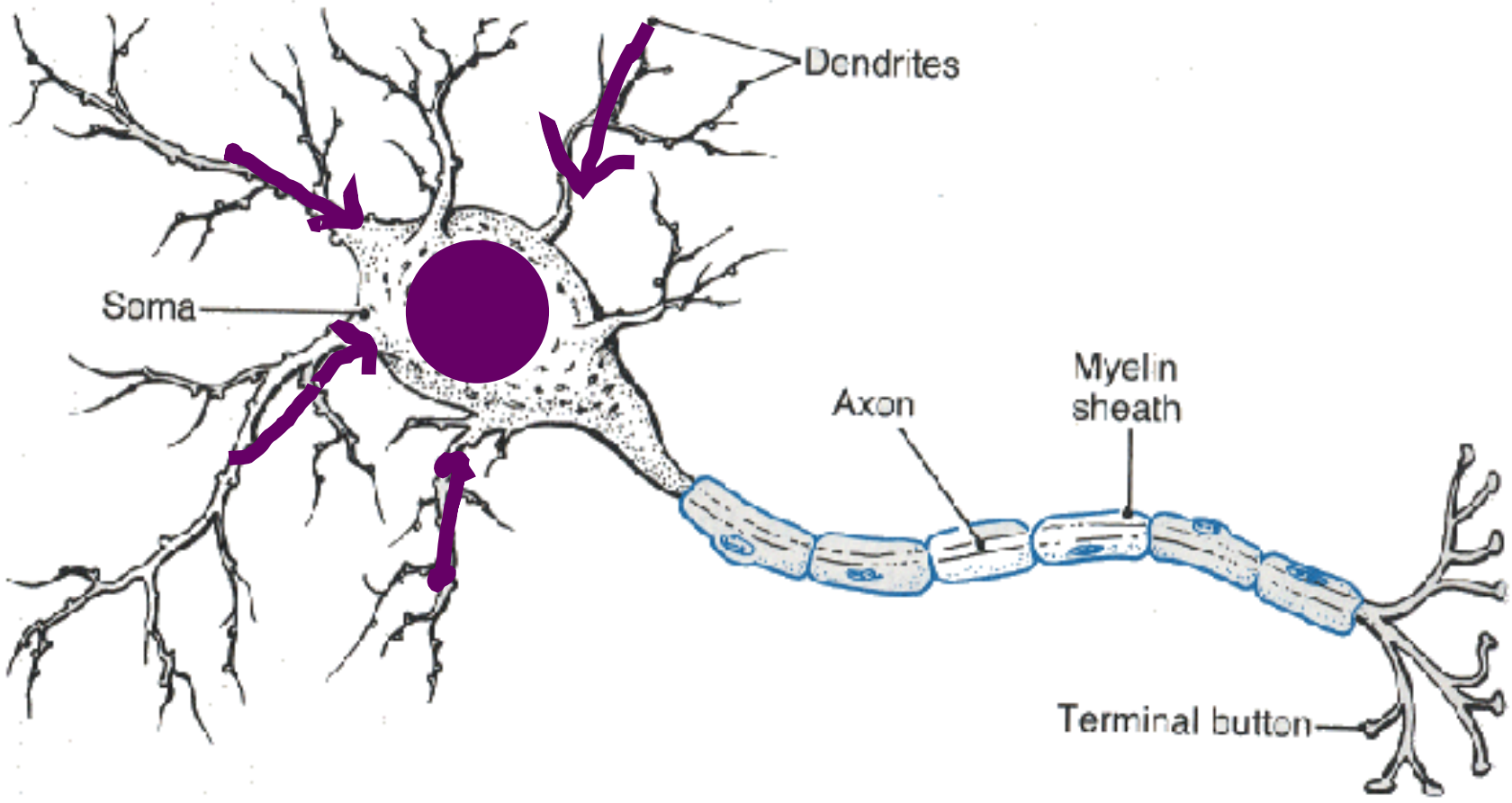
Neuron



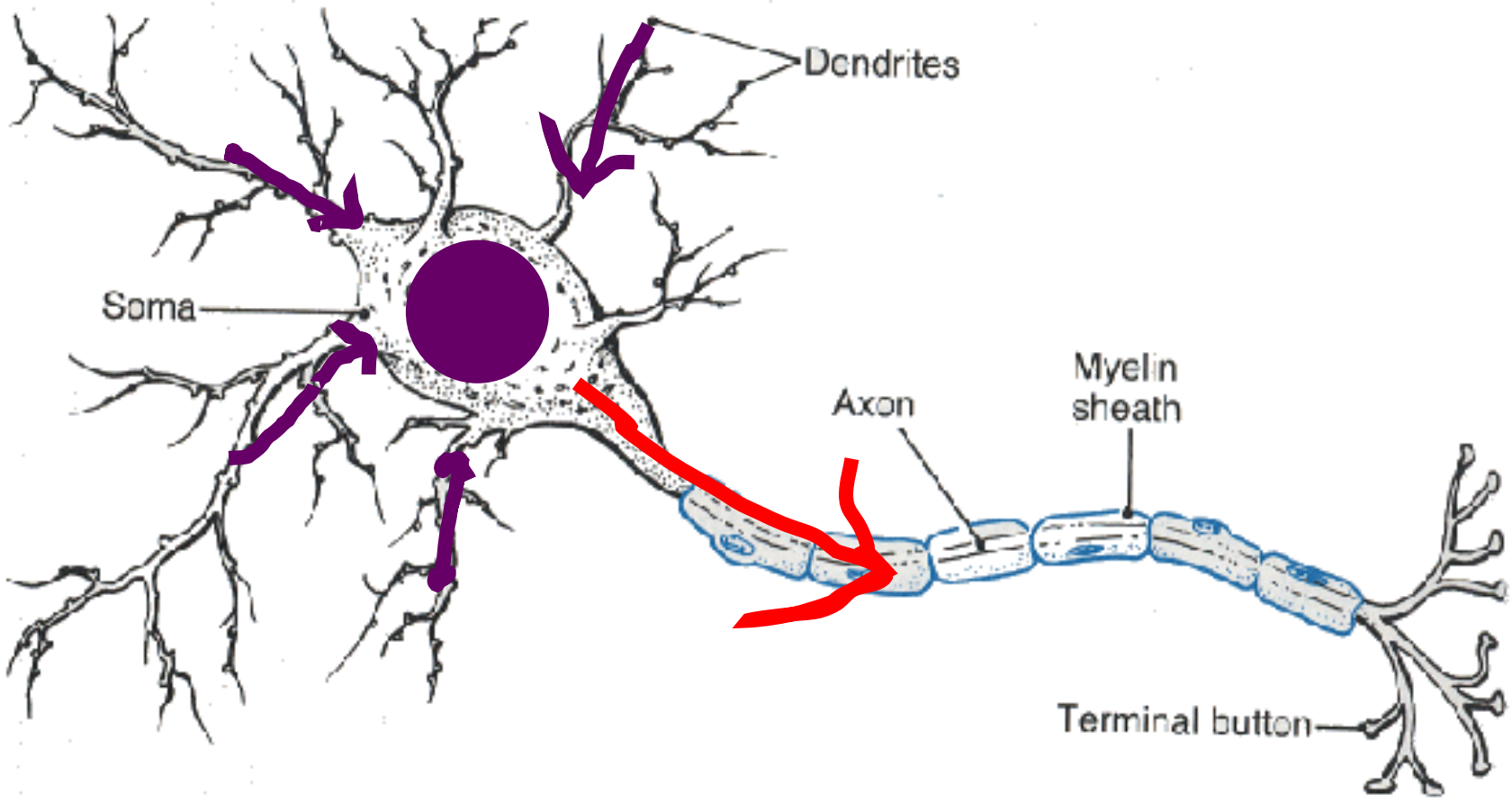
Neuron



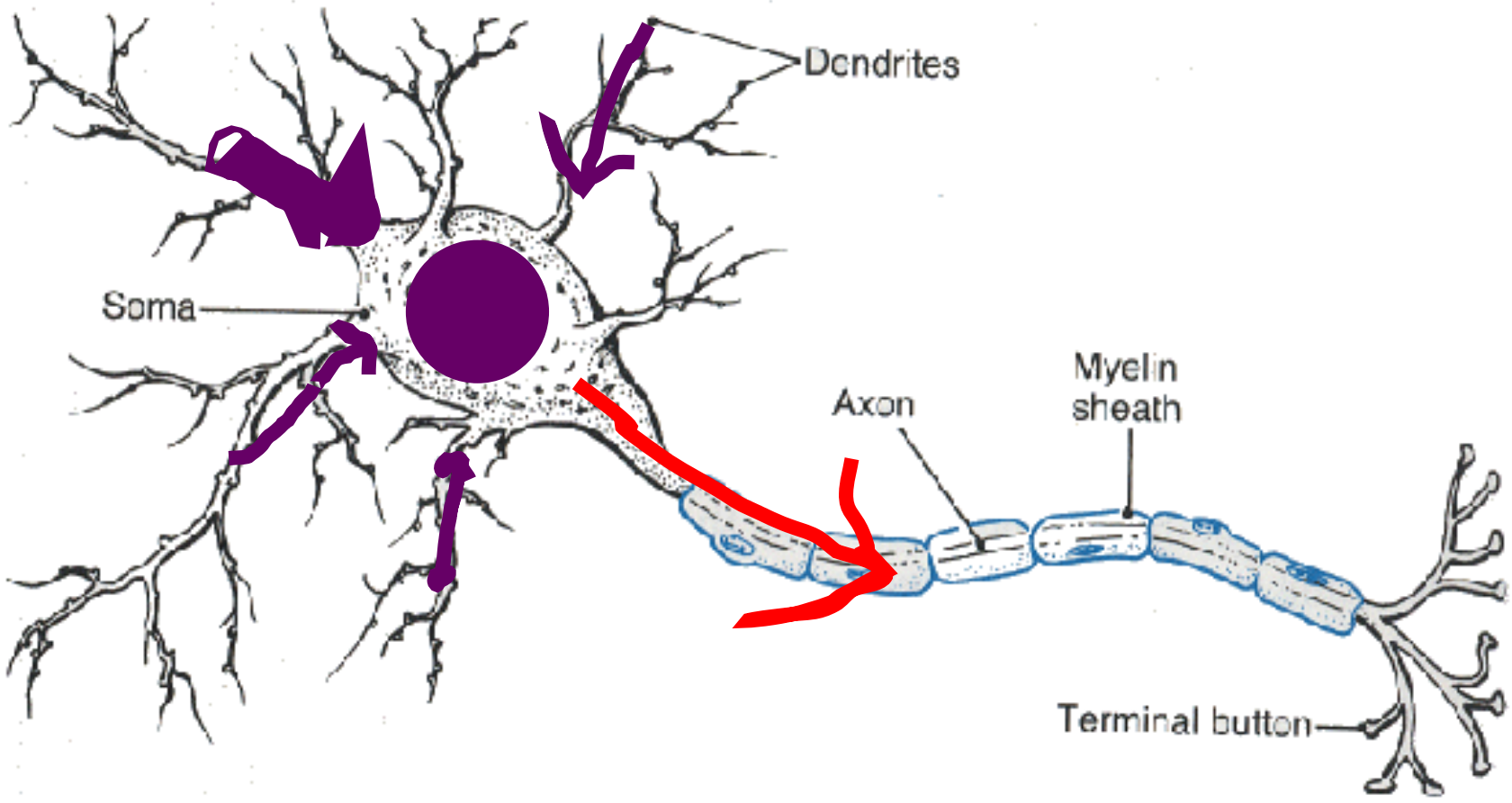
Neuron



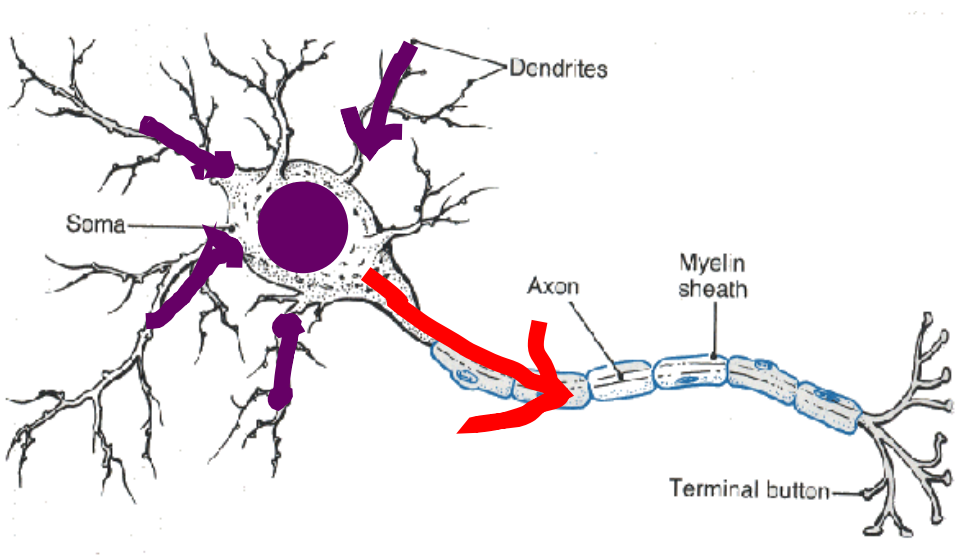
Neuron

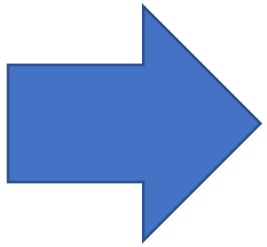


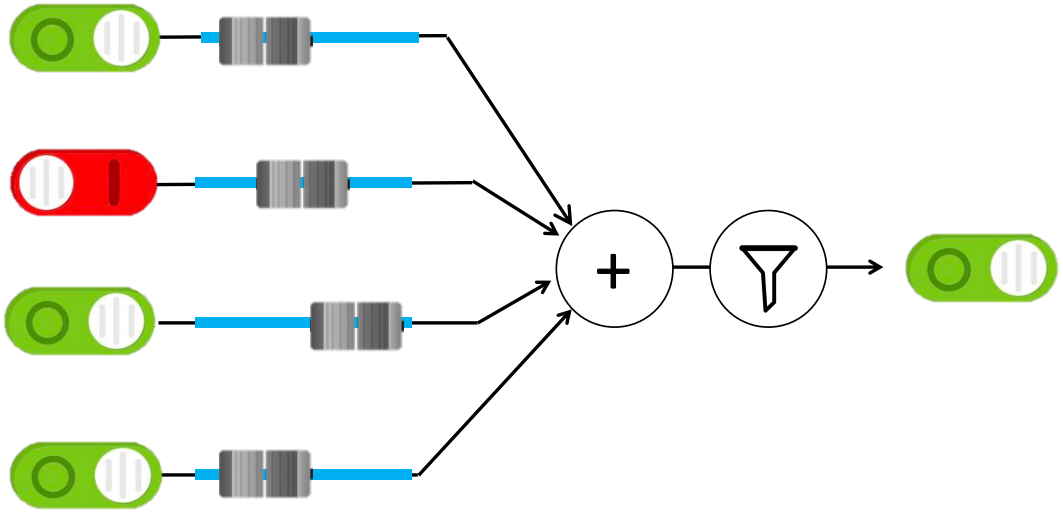
Some Inputs are More Important



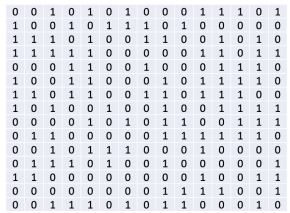
Artificial Neuron

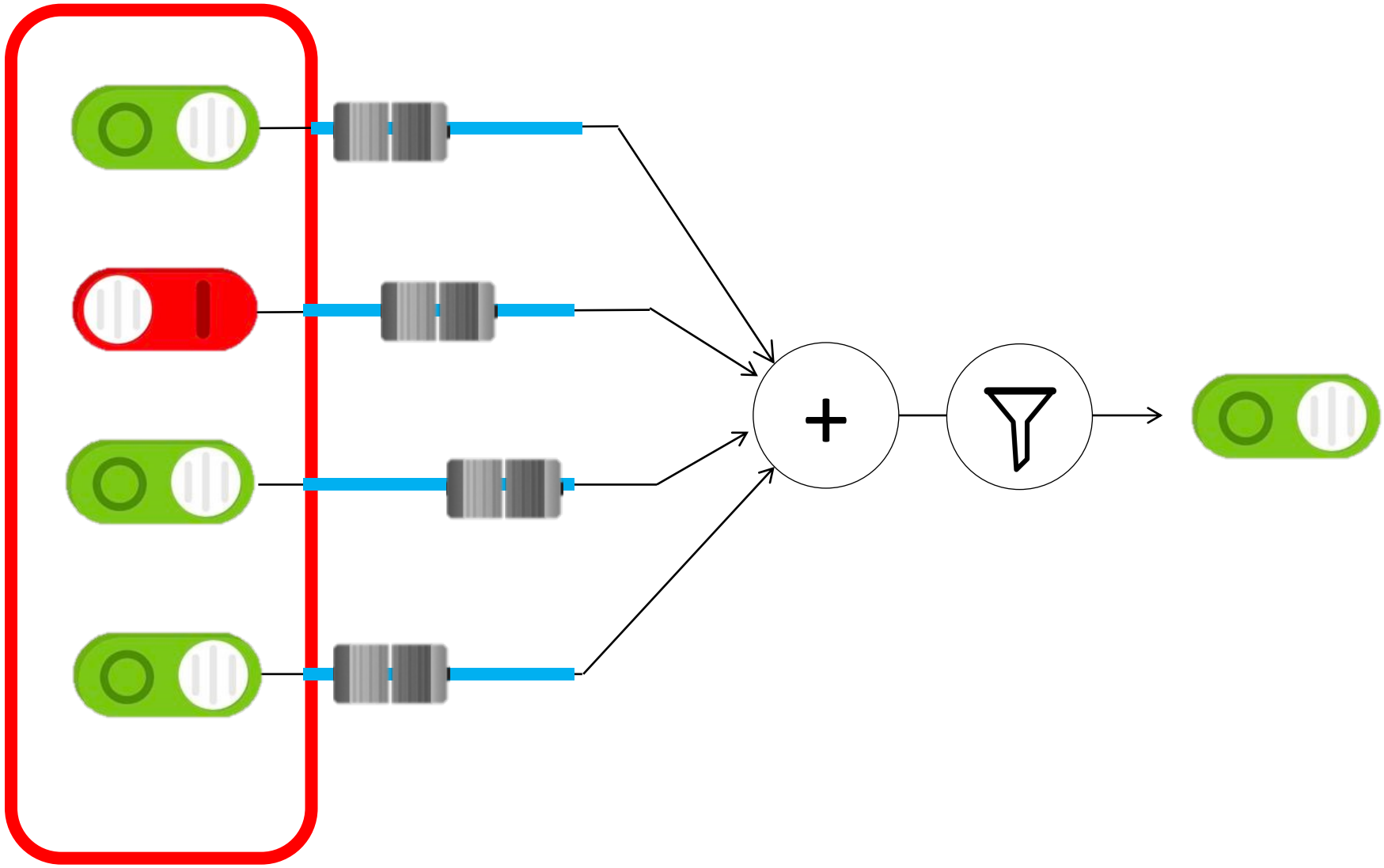




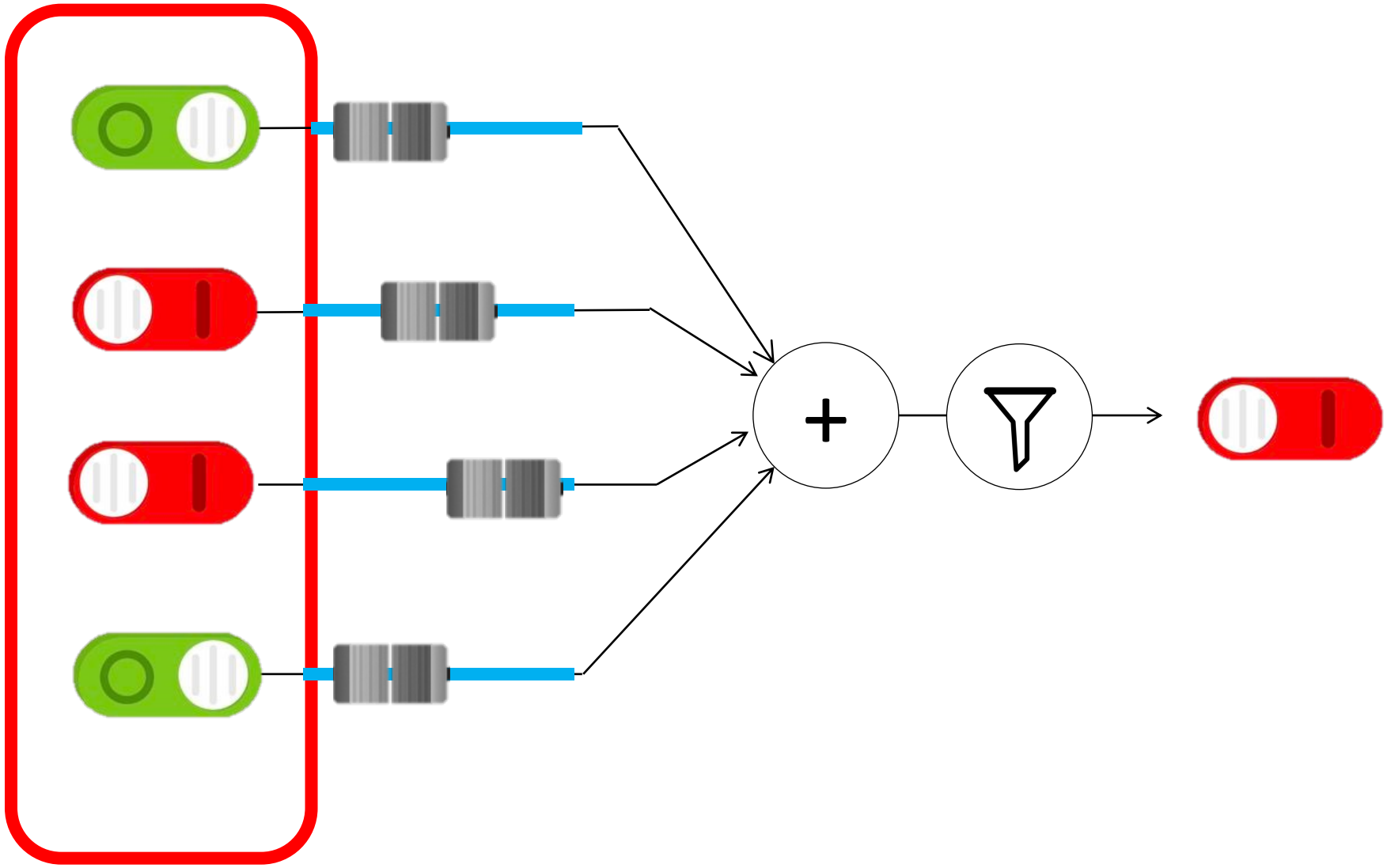


Artificial Neurons Have Inputs

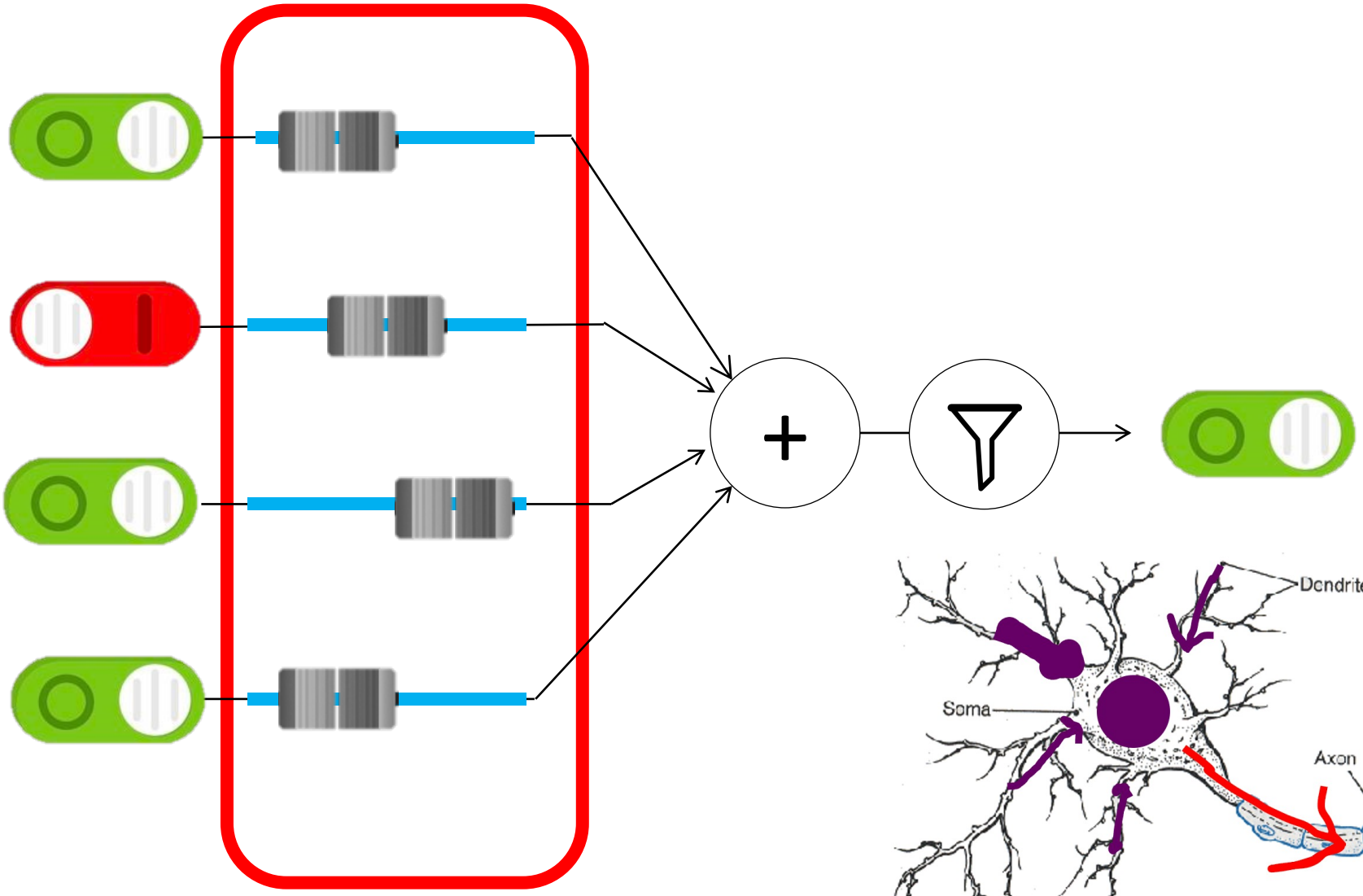




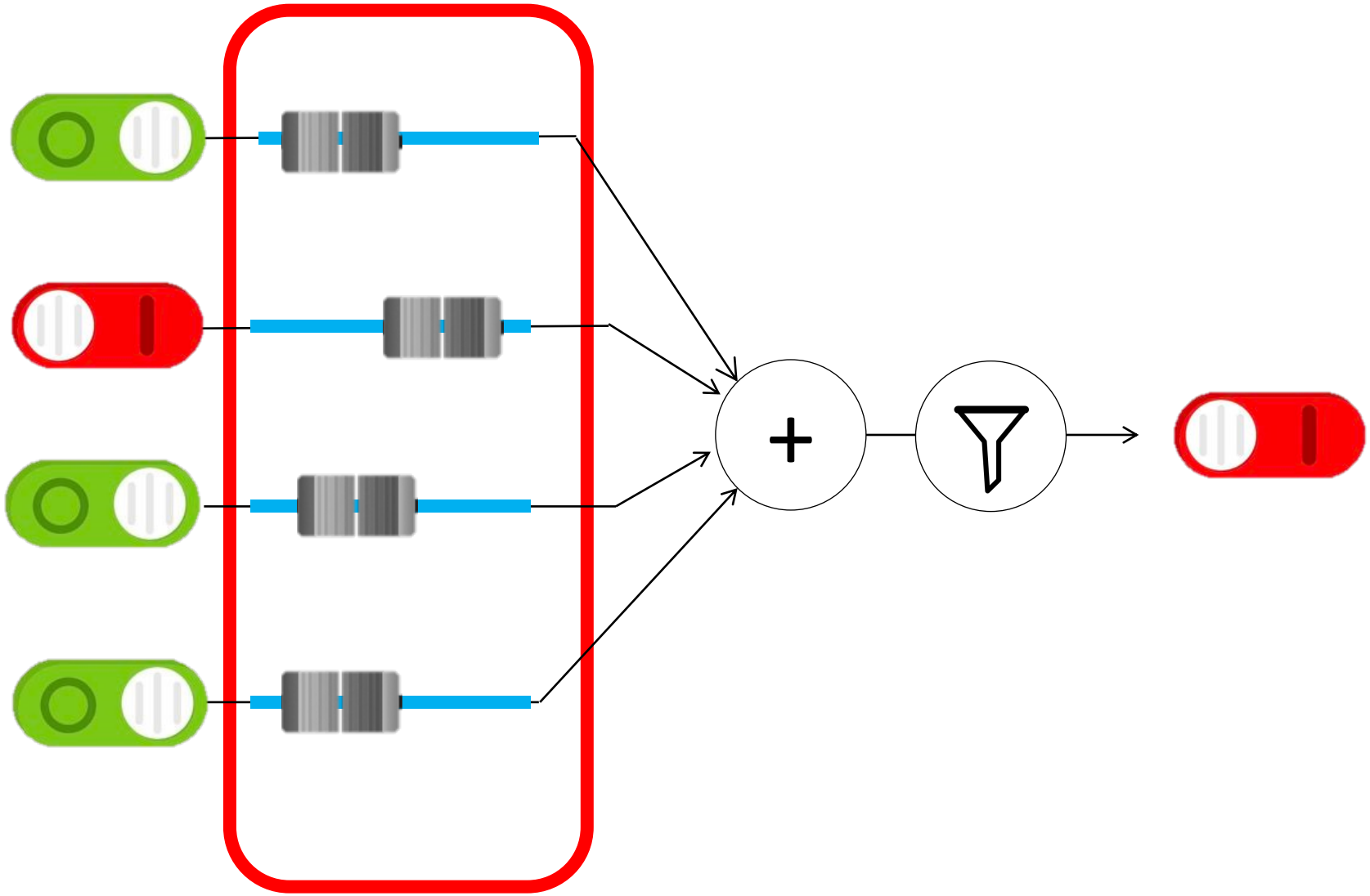
Different Inputs → Different Outputs



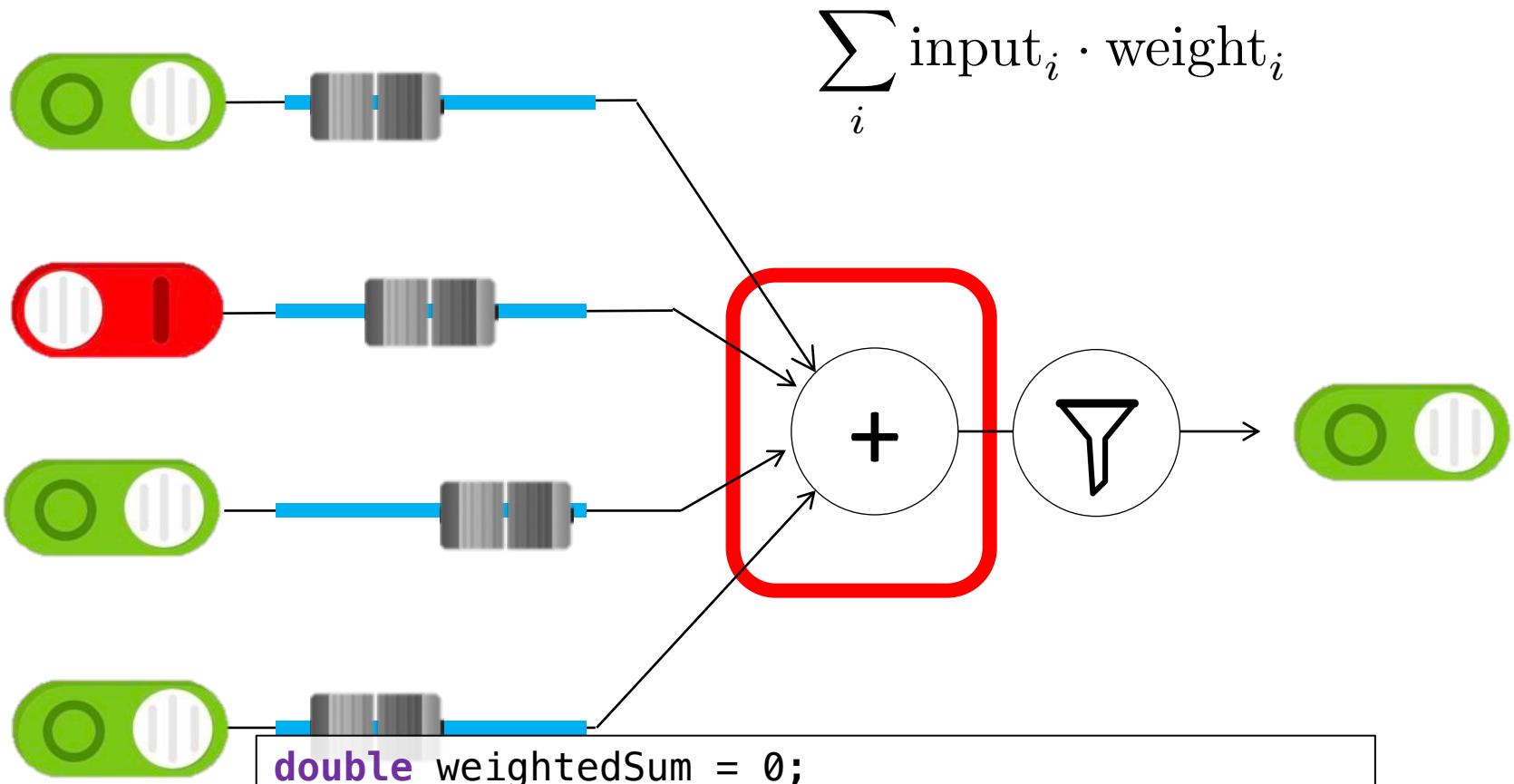
Different Weights → Different Outputs



Different Weights → Different Outputs



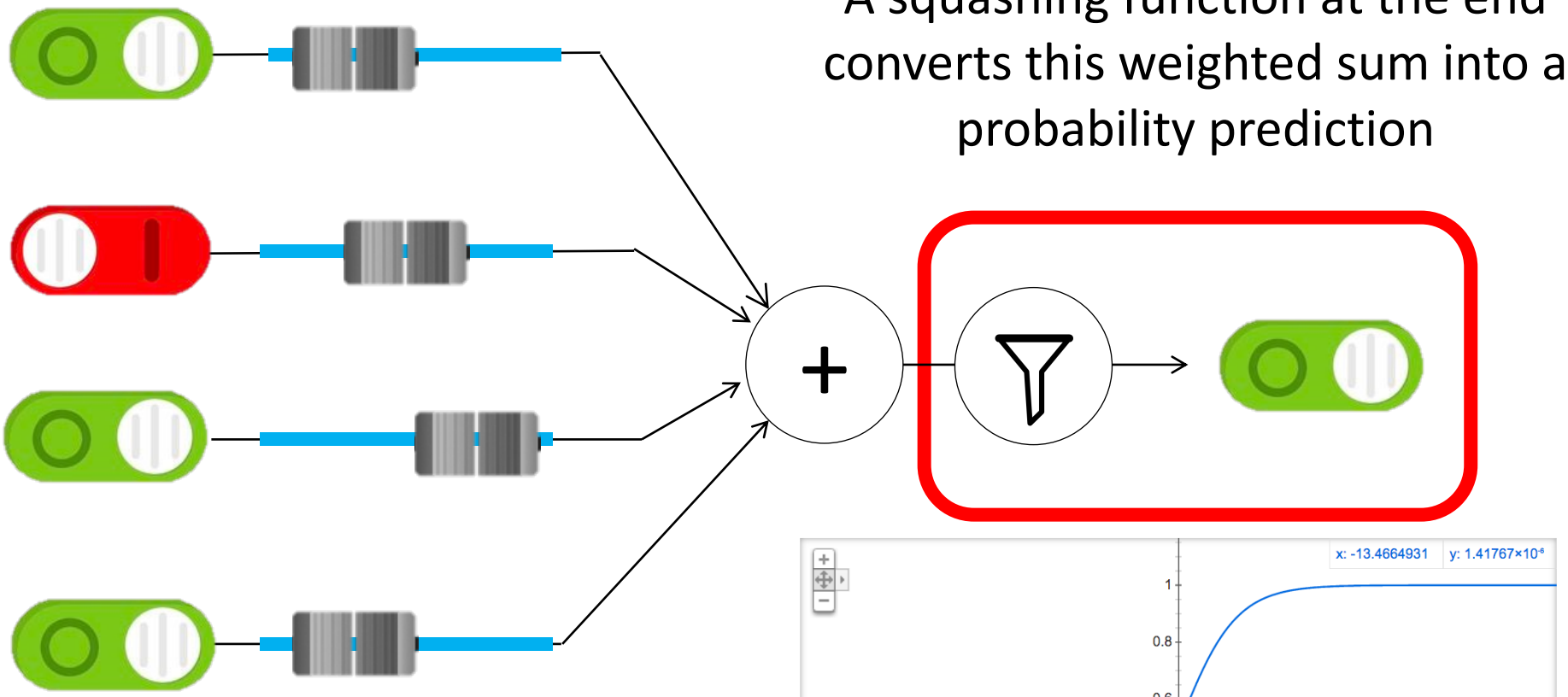
Computing A Weighted Sum Of Inputs

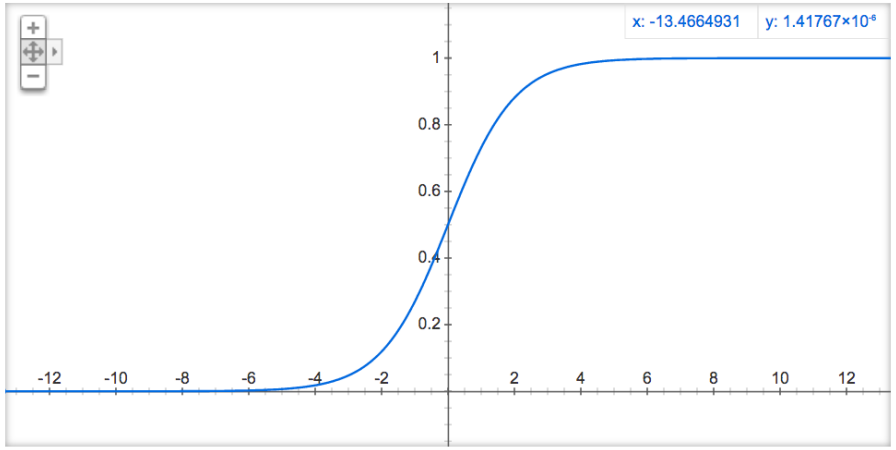



```
double weightedSum = 0;
weightedSum += input0 * weight0;
weightedSum += input1 * weight1;
weightedSum += input2 * weight2;
weightedSum += input3 * weight3;
```

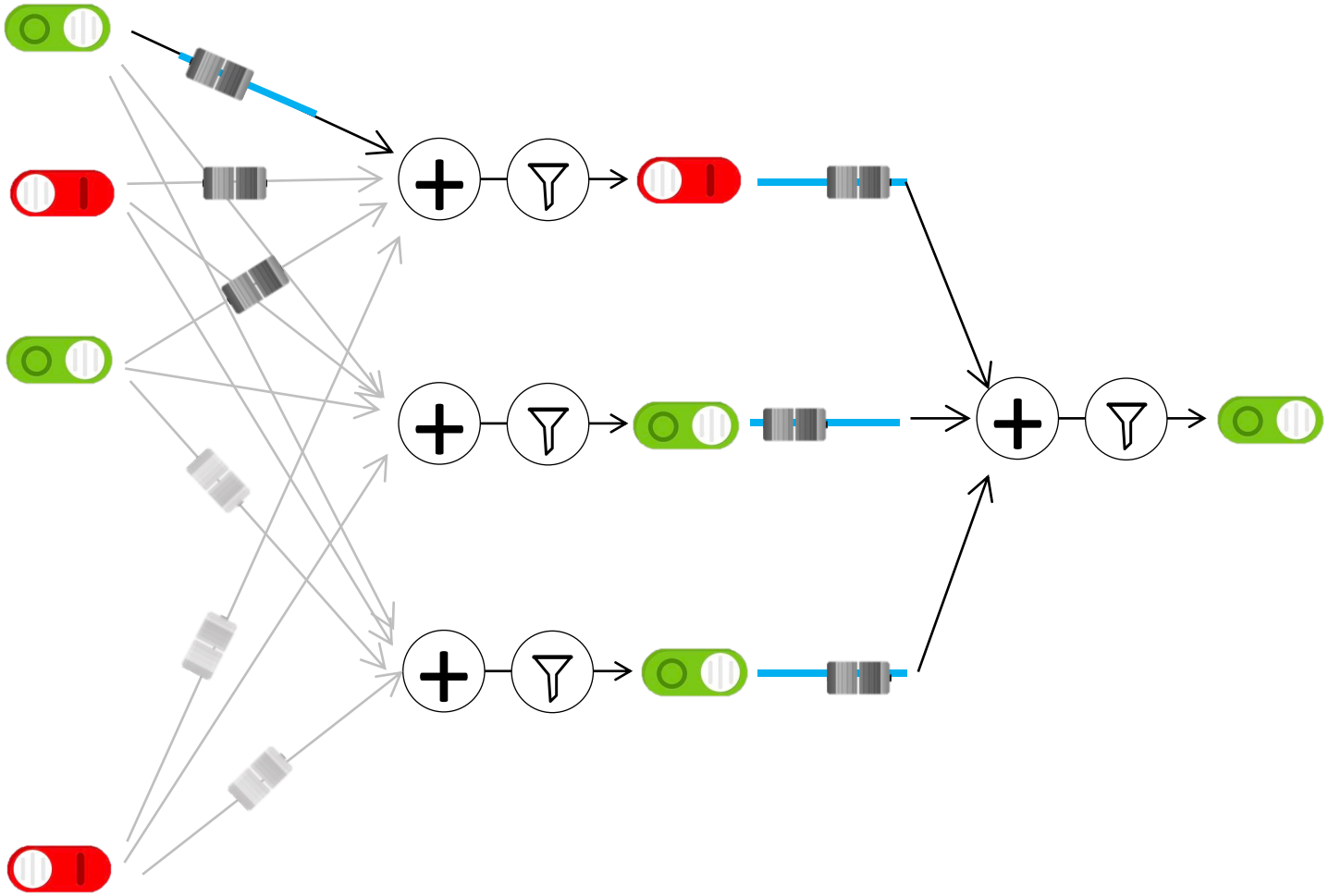
Filter and Output

A squashing function at the end converts this weighted sum into a probability prediction

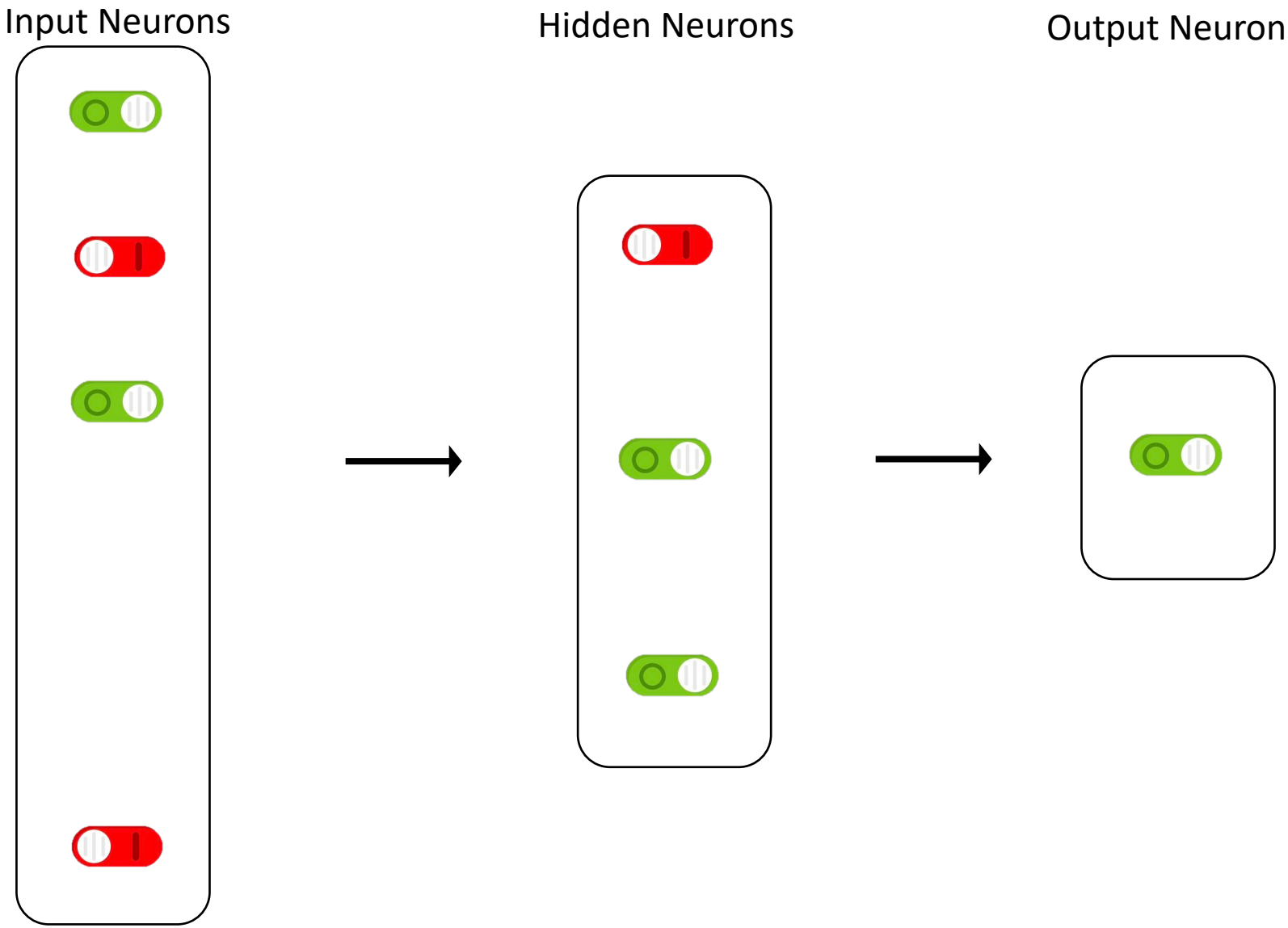




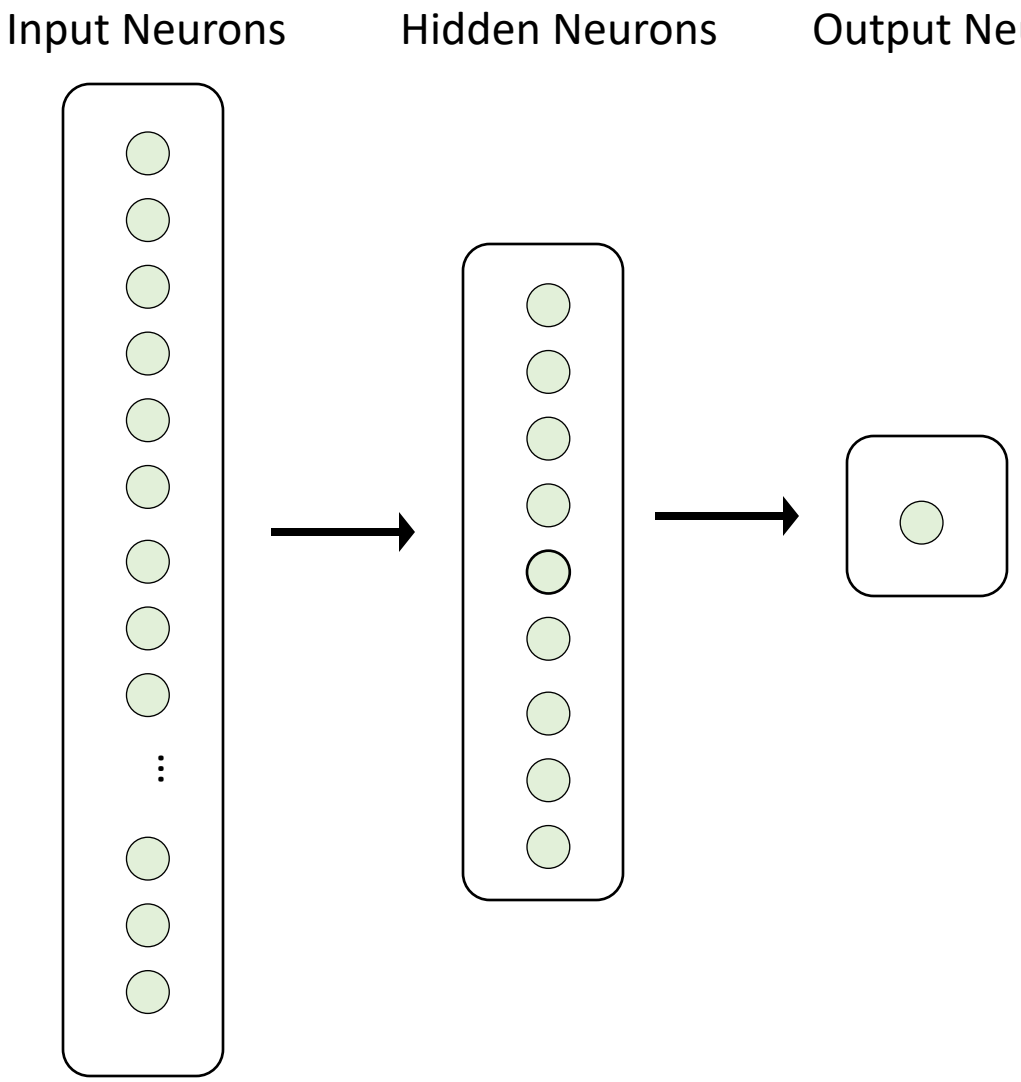
What If We Put Many Together?



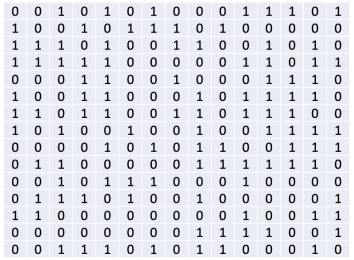
What If We Put Many Together?



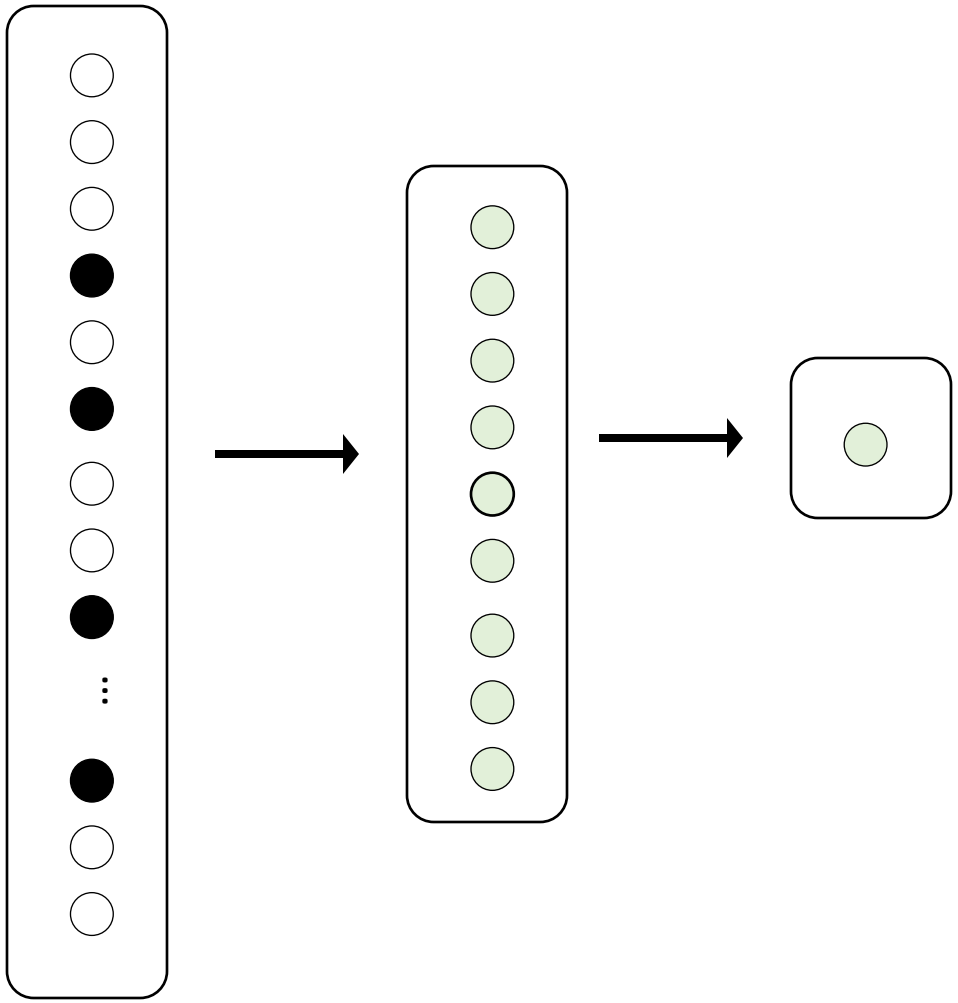
Making a Prediction



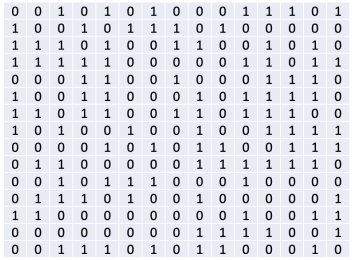
Making a Prediction

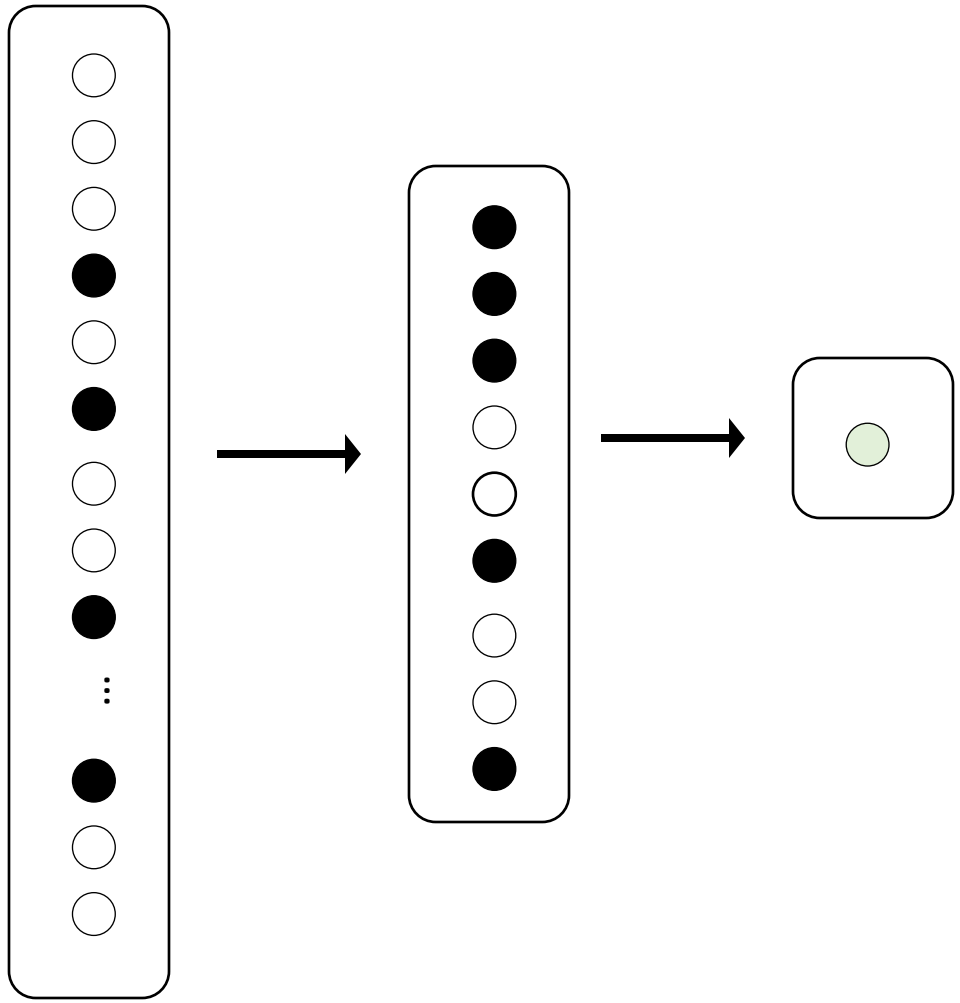


Input Neurons Hidden Neurons Output Neurons

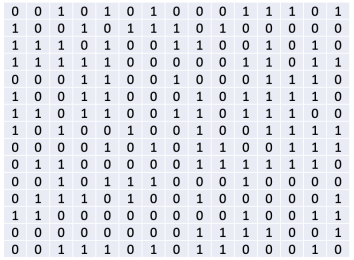


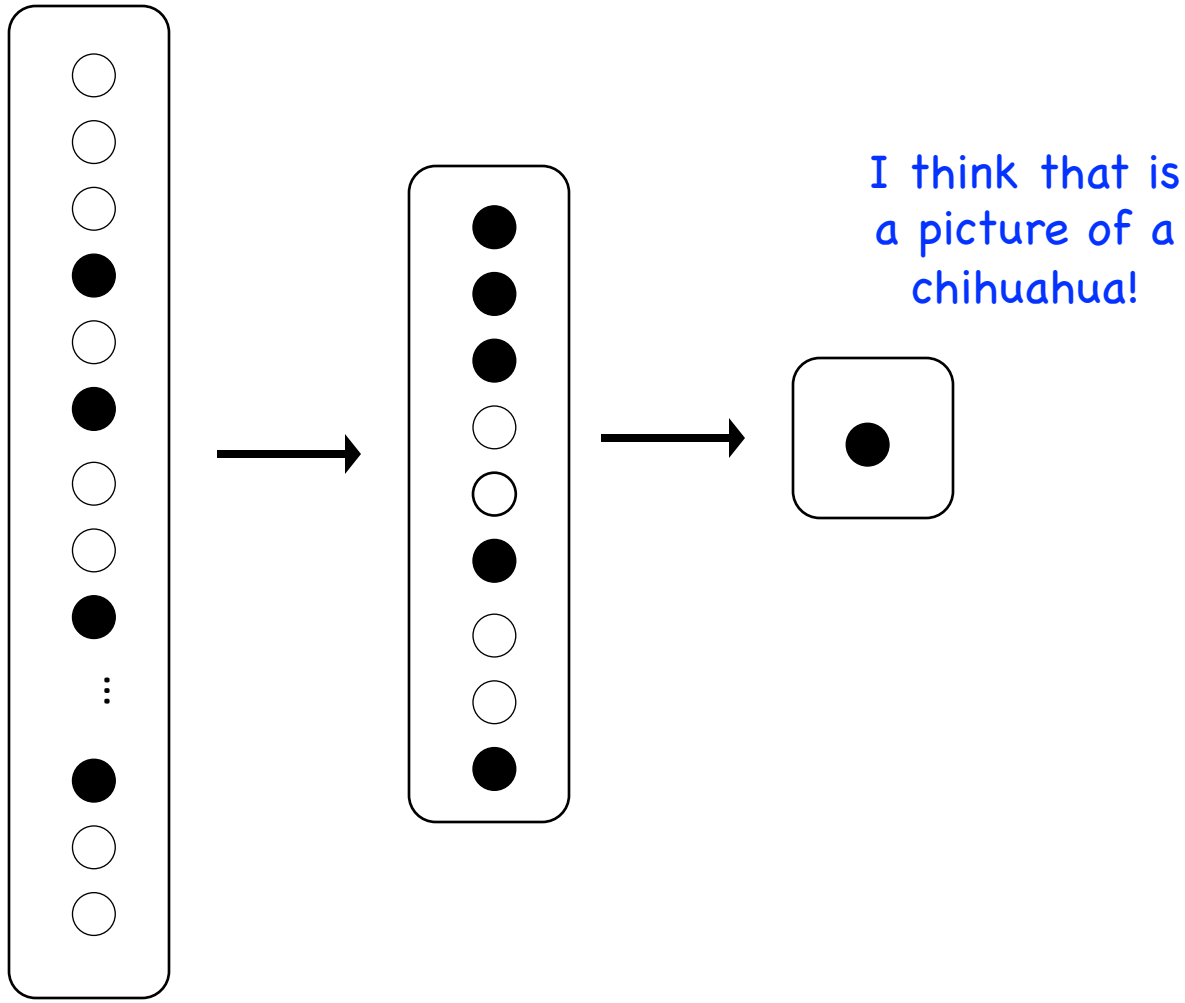
Making a Prediction



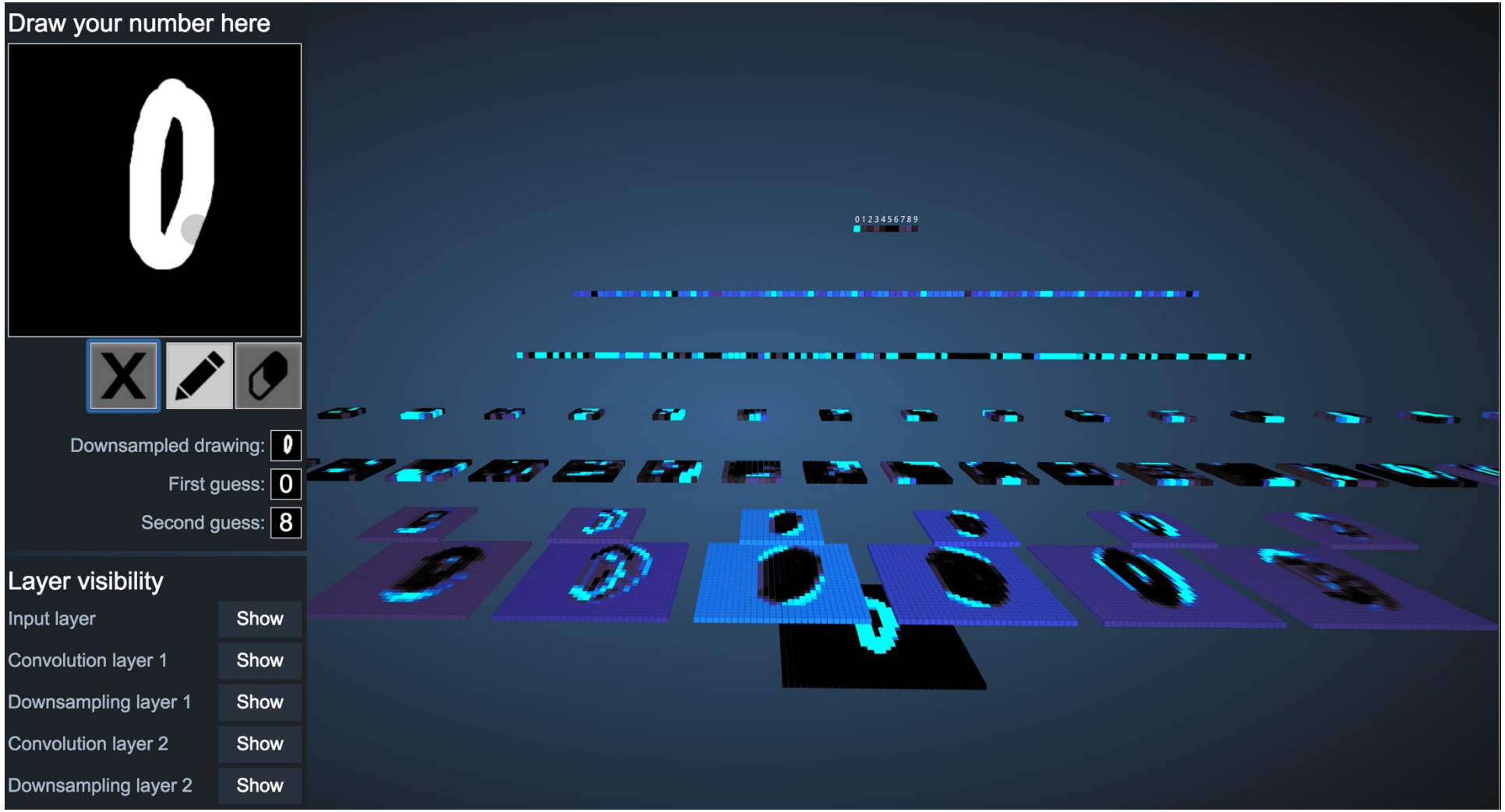


Making a Prediction





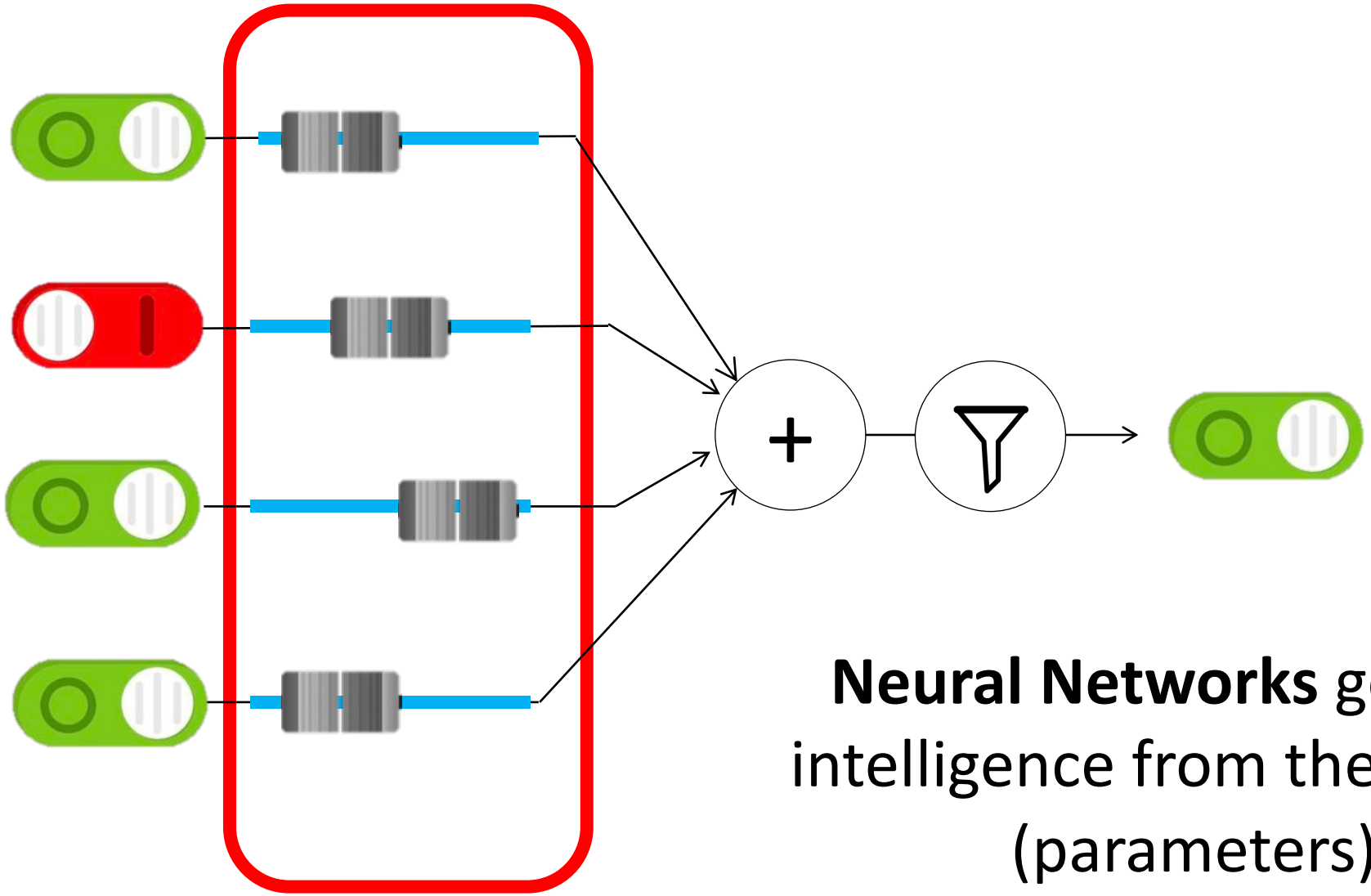
Demonstration



https://web.archive.org/web/20211117115916/https://www.cs.ryerson.ca/~aharley/vis/conv/

Where do Artificial Neural Networks get their intelligence from?

Intelligence Is In The Weights



Neural Networks get their intelligence from their sliders (parameters).

Two Great Ideas

1. Artificial Neurons

2. Learn by Example

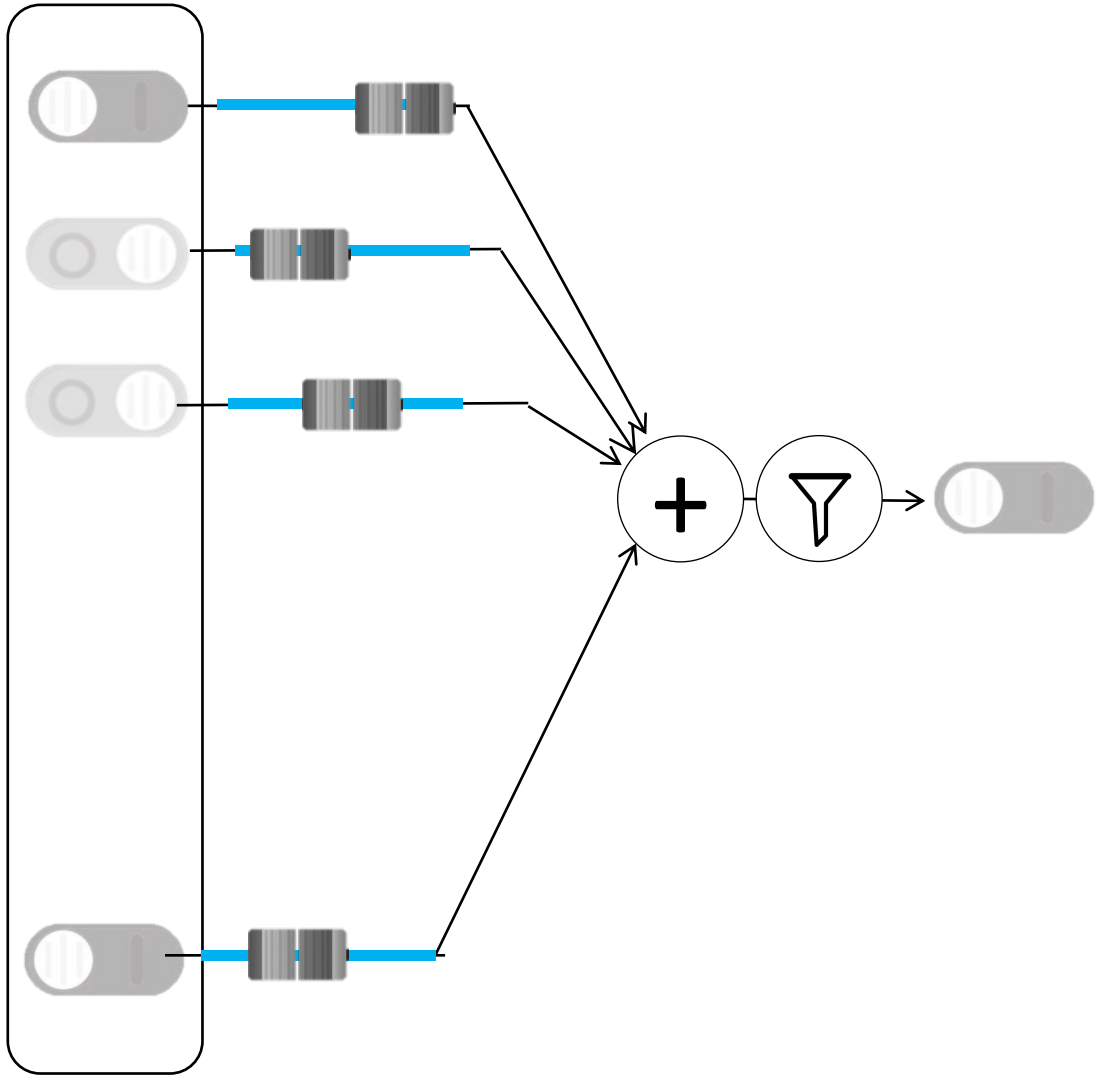
Two Great Ideas

1. Artificial Neurons

2. Learn by Example

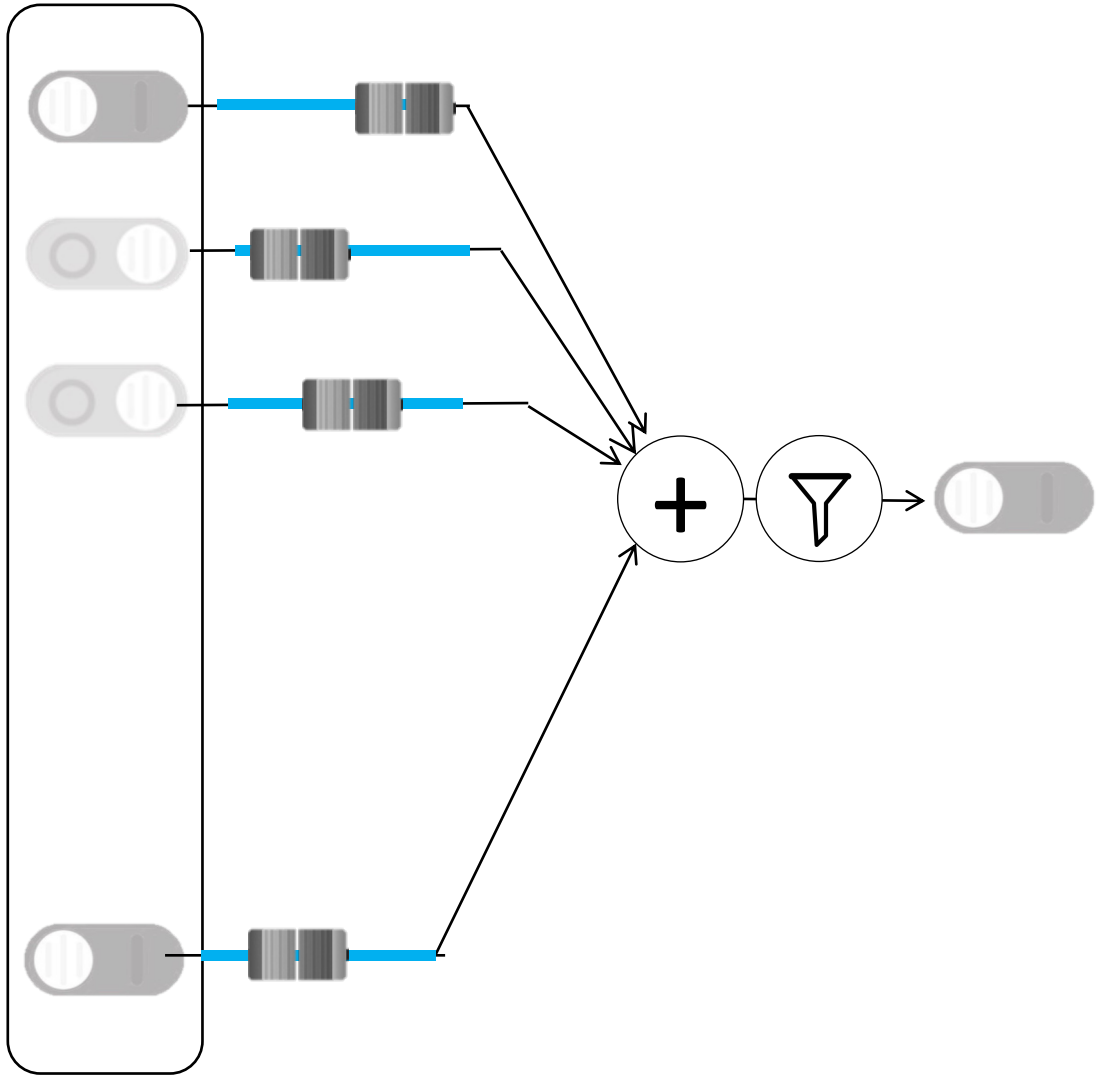
Learn by Example

Learn by Example



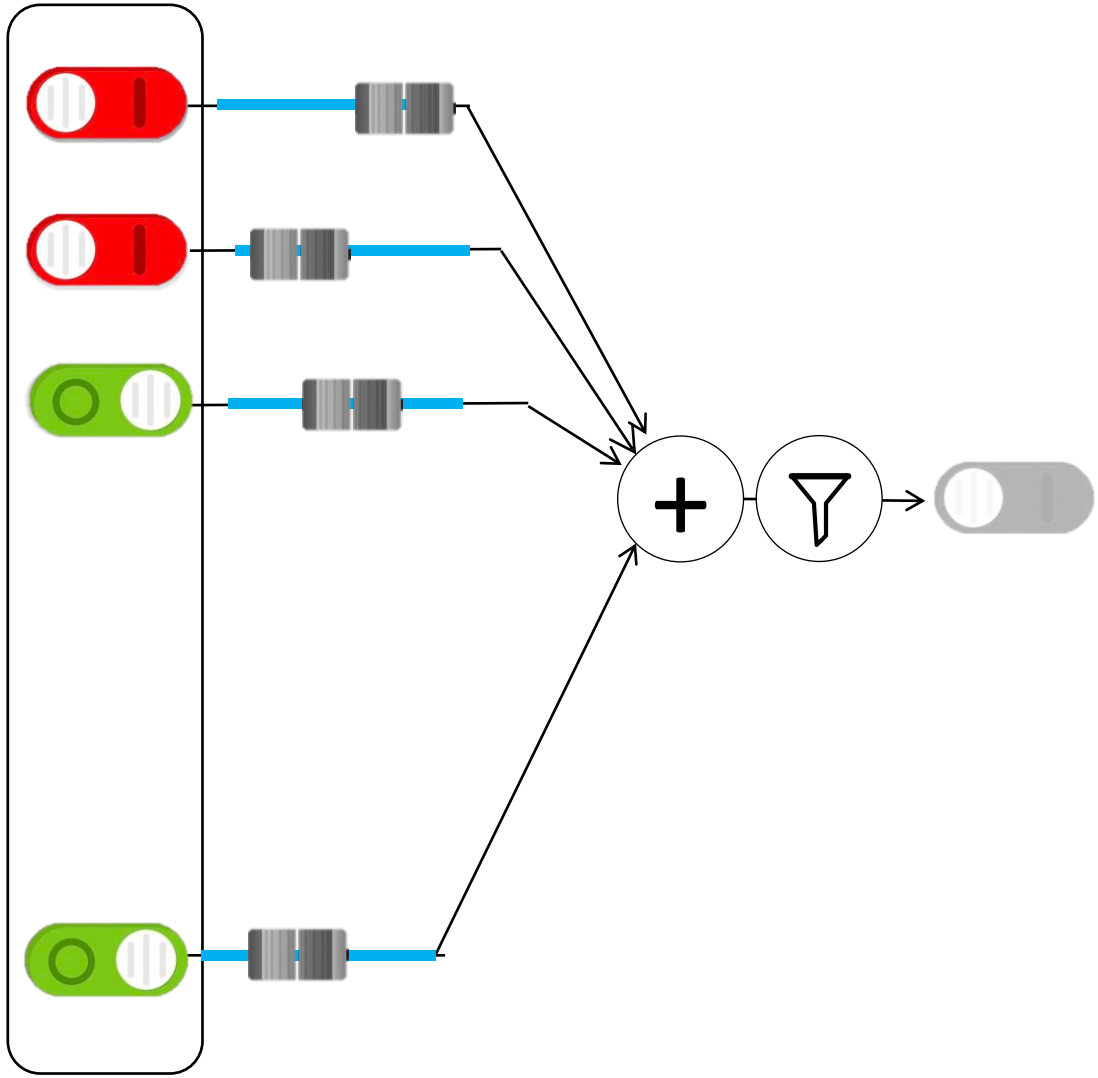
Learn by Example

Here is an example for the neuron to train on.



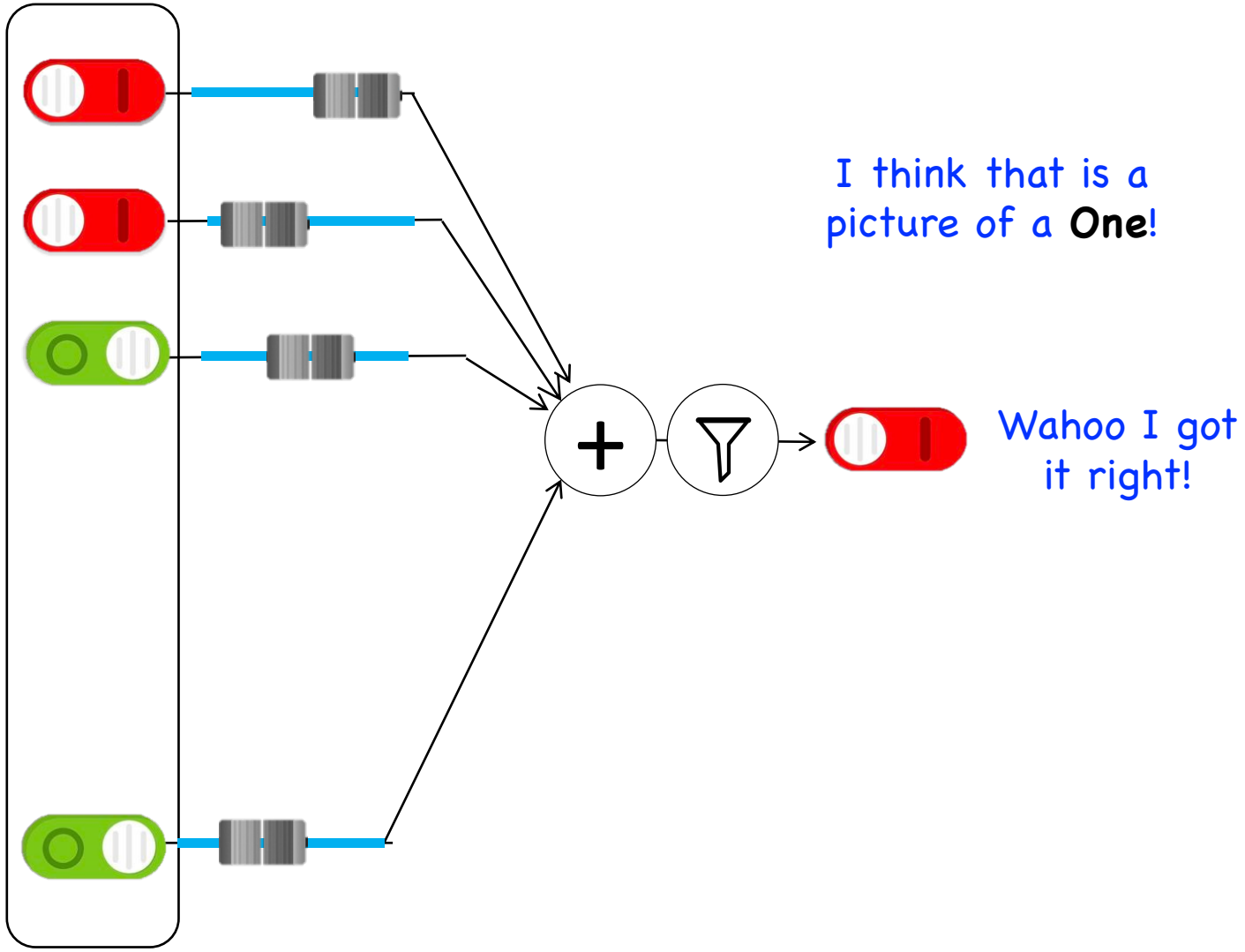
Learn by Example

Here is an example for the neuron to train on.



Learn by Example

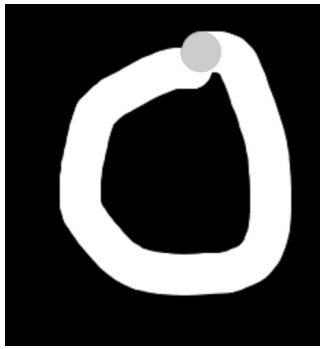
Here is an example for the neuron to train on.



I think that is a picture of a **One**!

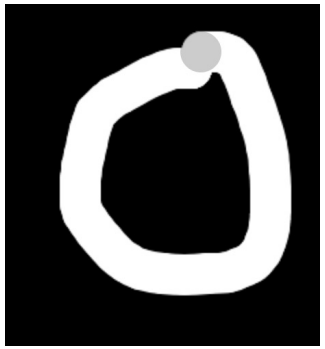
Wahoo I got it right!

Learn by Example

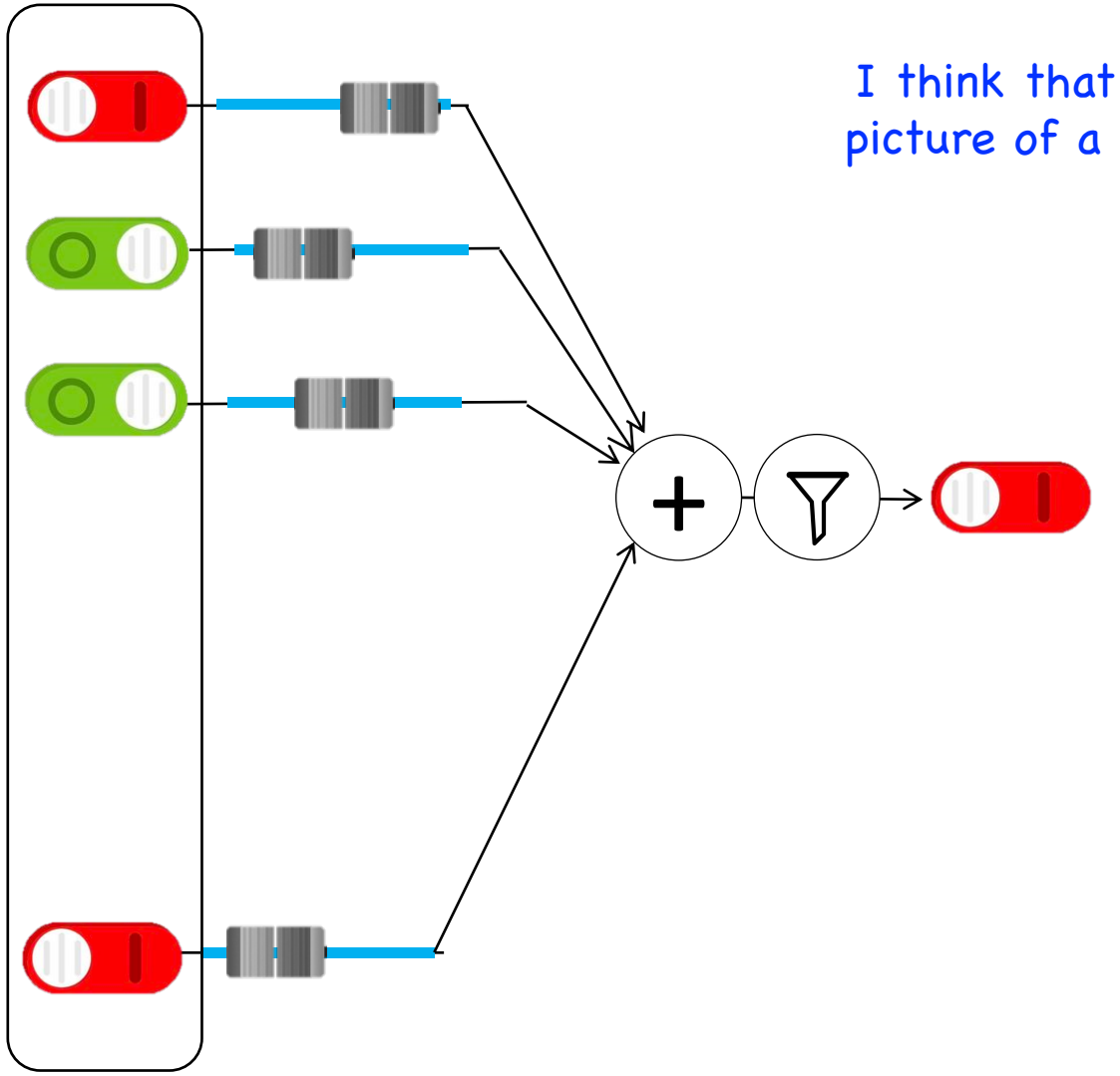


Here is another training example.

Learn by Example

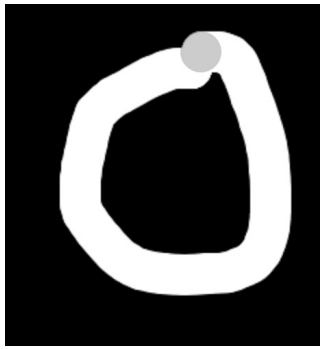


Here is another training example.

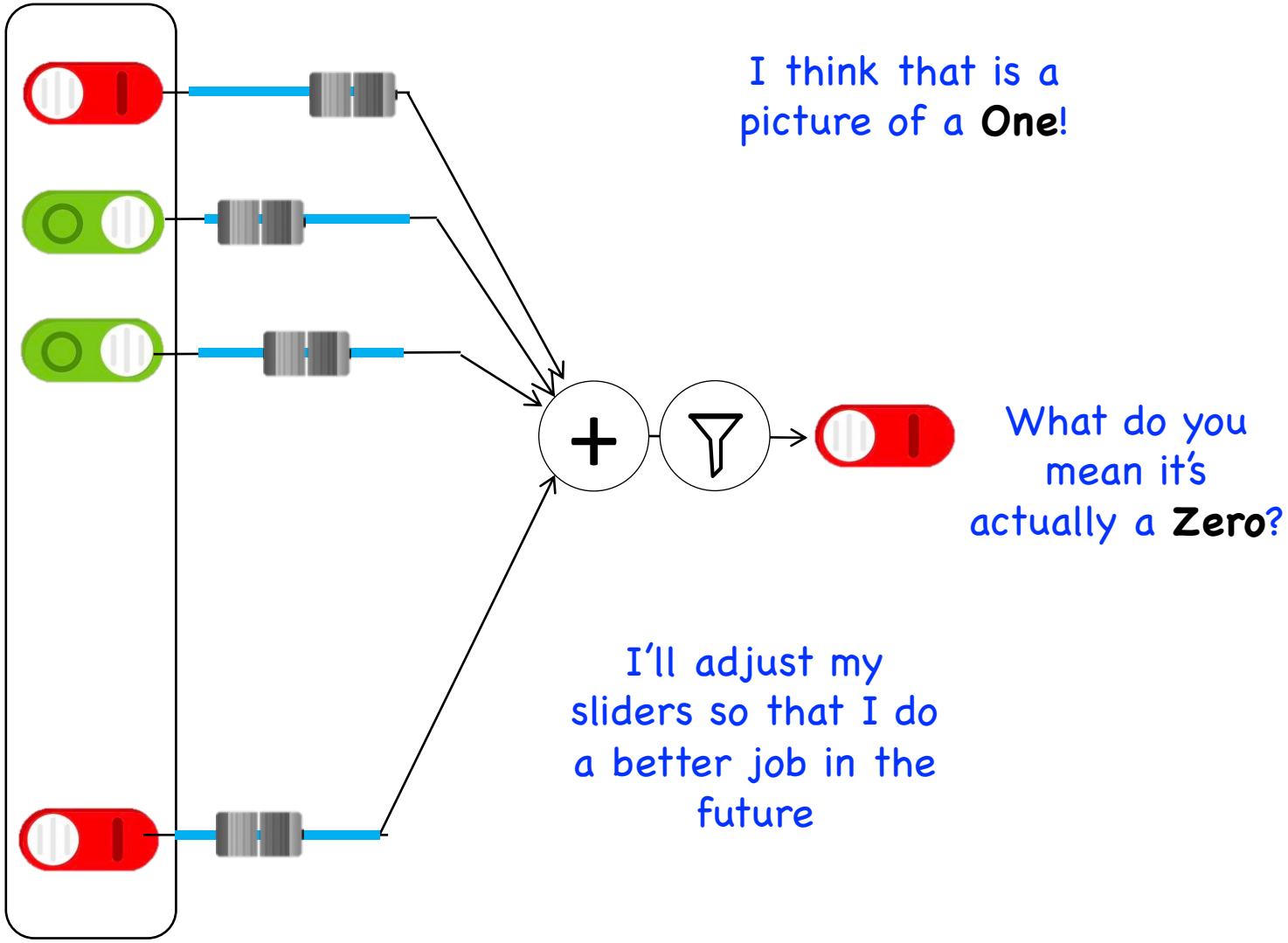


I think that is a picture of a **One**!

Learn by Example



Here is another training example.

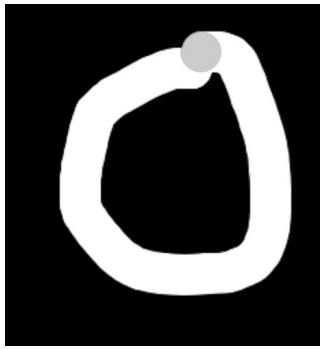


I think that is a picture of a **One**!

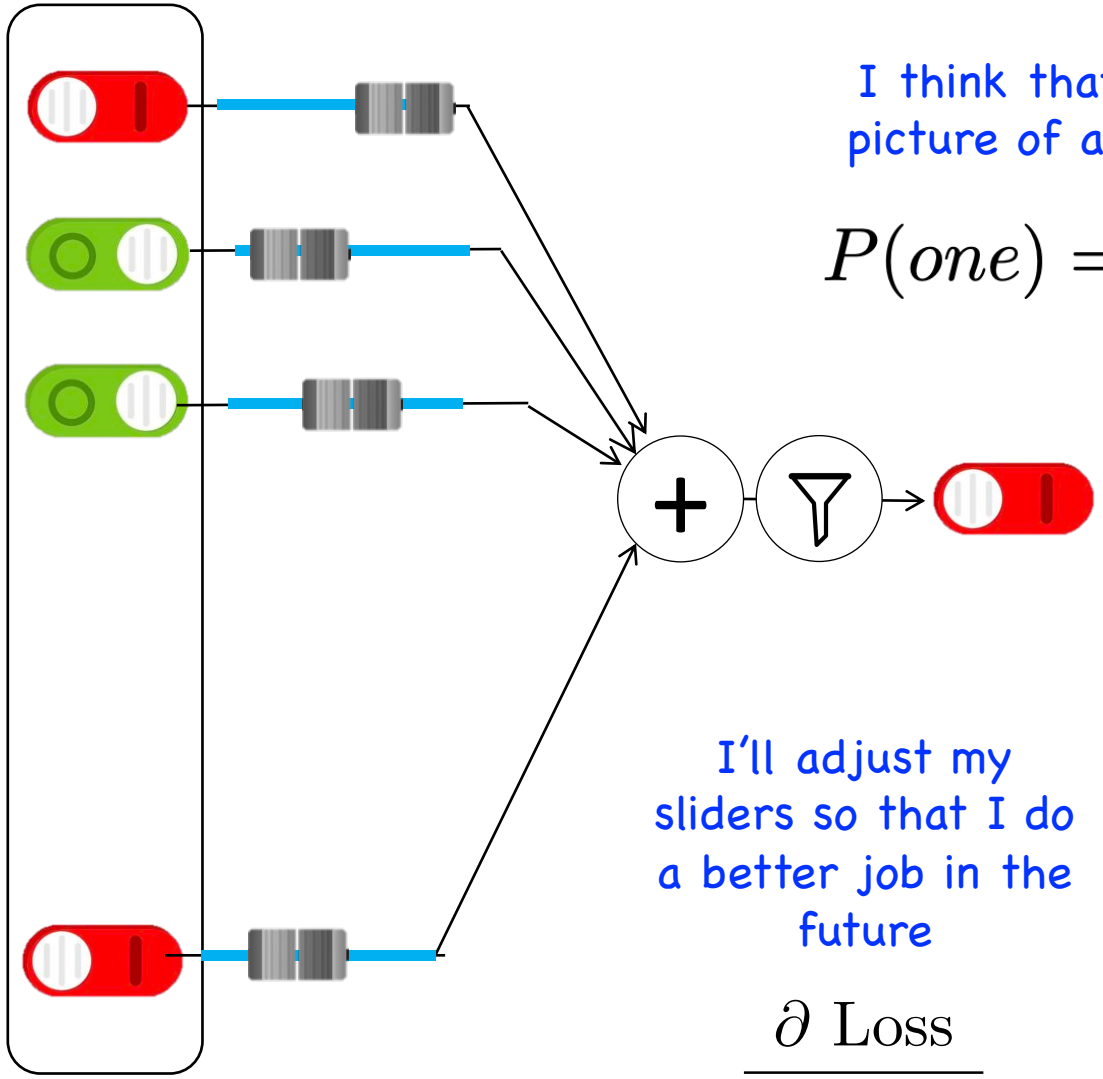
What do you mean it's actually a **Zero**?

I'll adjust my sliders so that I do a better job in the future

Learn by Example



Here is another training example.



I think that is a picture of a **One**!

$$P(one) = 0.8$$

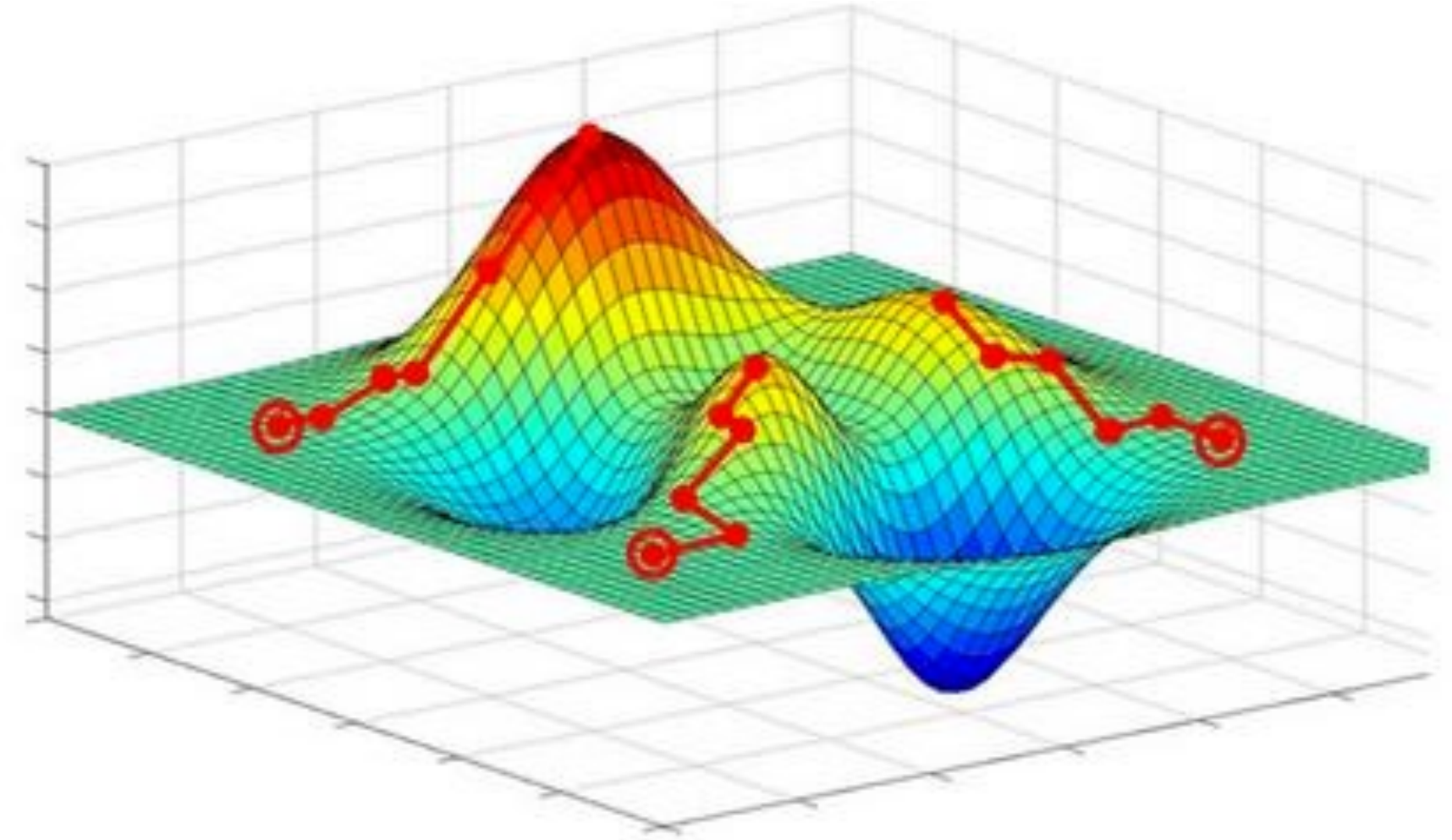
What do you mean it's actually a **Zero**?

$$\text{Loss} = 1$$

I'll adjust my sliders so that I do a better job in the future

$$\frac{\partial\ \text{Loss}}{\partial\ \text{Slider}_i}$$

Neurons Learn via Gradients



Walk uphill and you will find a local maxima
(if your step size is small enough)

Piech & Cain, CS109, Stanford University

Neurons Learn via Gradients of Probability

$$\frac{\partial L}{\partial \theta_i^{(\hat{y})}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i^{(\hat{y})}}$$

$$\hat{y} = \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)$$

$$\frac{\partial \hat{y}}{\partial \theta_i^{(\hat{y})}} = \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)\left[1 - \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)\right] \cdot \frac{\partial}{\partial \theta_i^{(\hat{y})}} \sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}$$

$$= \hat{y}[1 - \hat{y}] \cdot \frac{\partial}{\partial \theta_i^{(\hat{y})}} \sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}$$

$$= \hat{y}[1 - \hat{y}] \cdot h_i$$

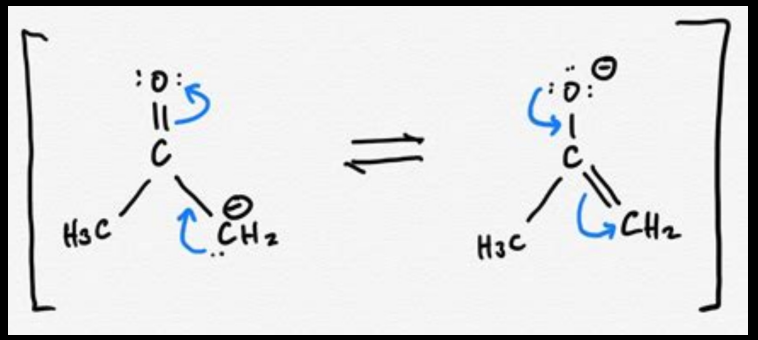
You will be able to do this by the end of class!

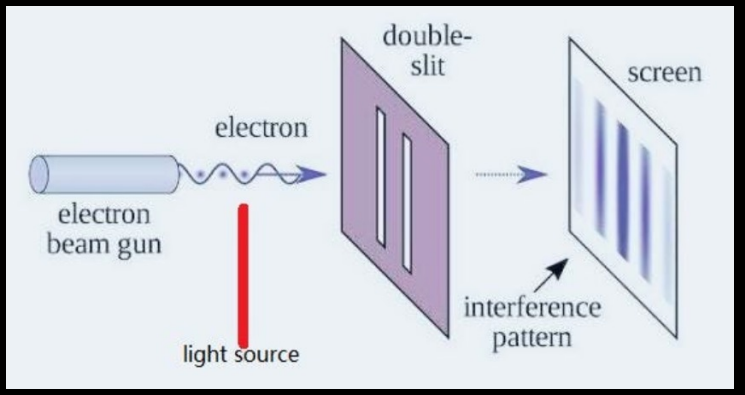
AI has constantly been revolutionized by people who understood probability theory.

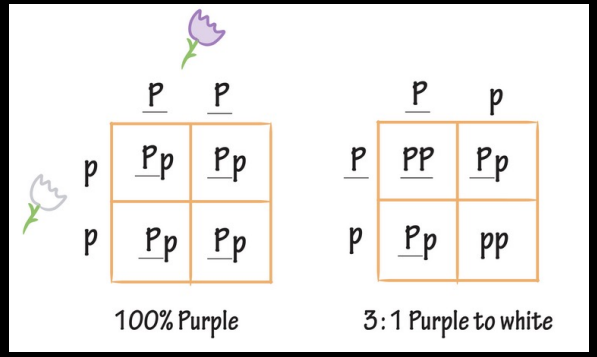
End of Story

Except it isn’t the end of the story…

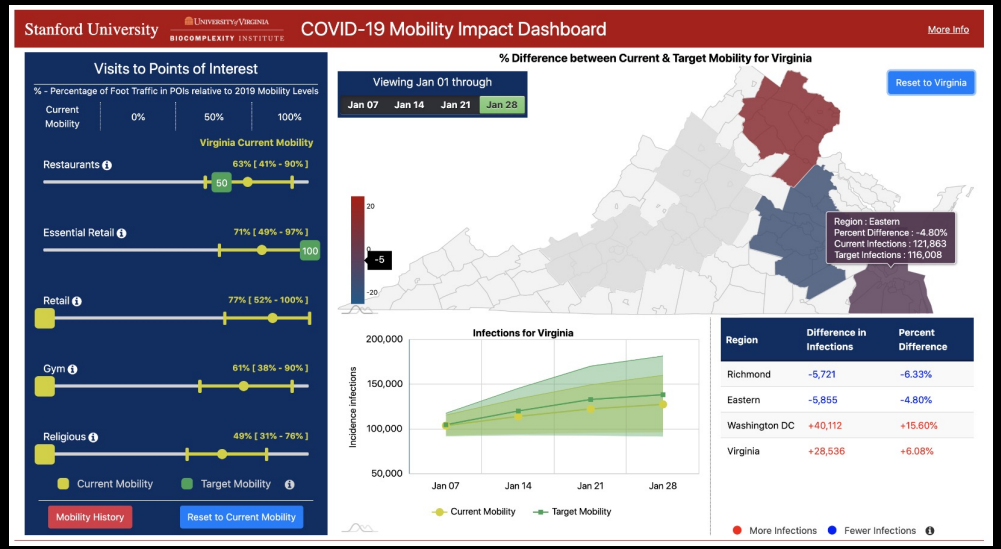
Probability is WAY more than just machine learning

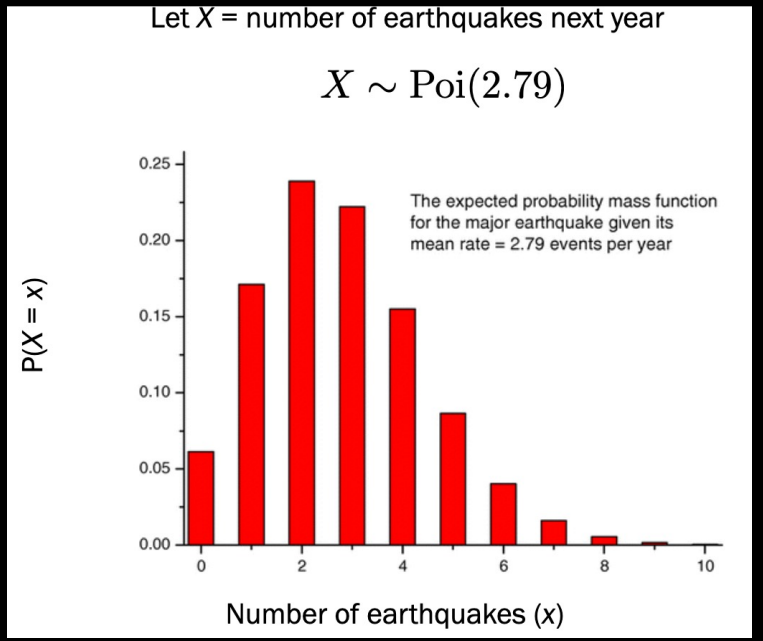






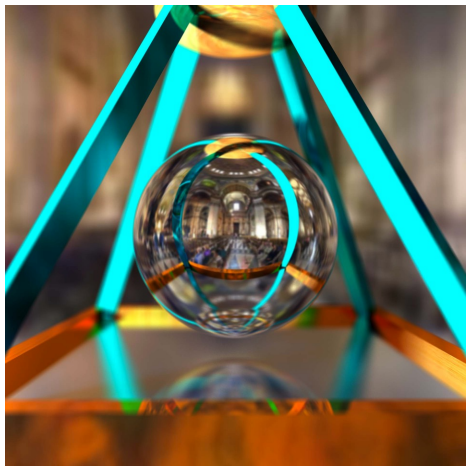
Probability is Everywhere



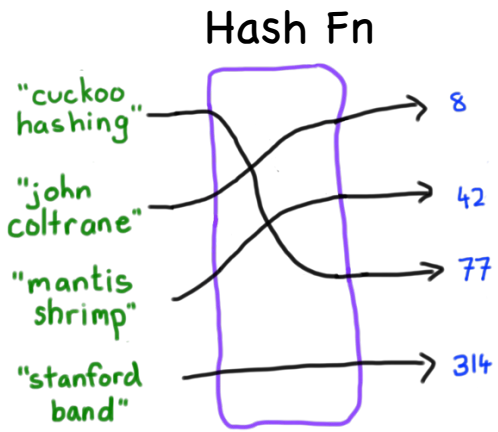


So Much "Pure CS" Relies on Probability

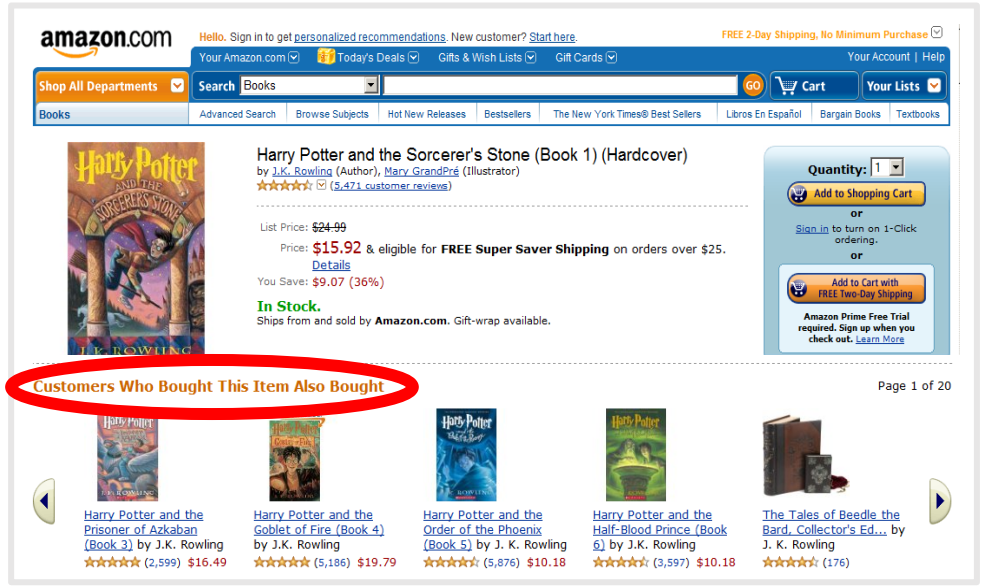
Raytracing



HashMaps



Recommender Systems

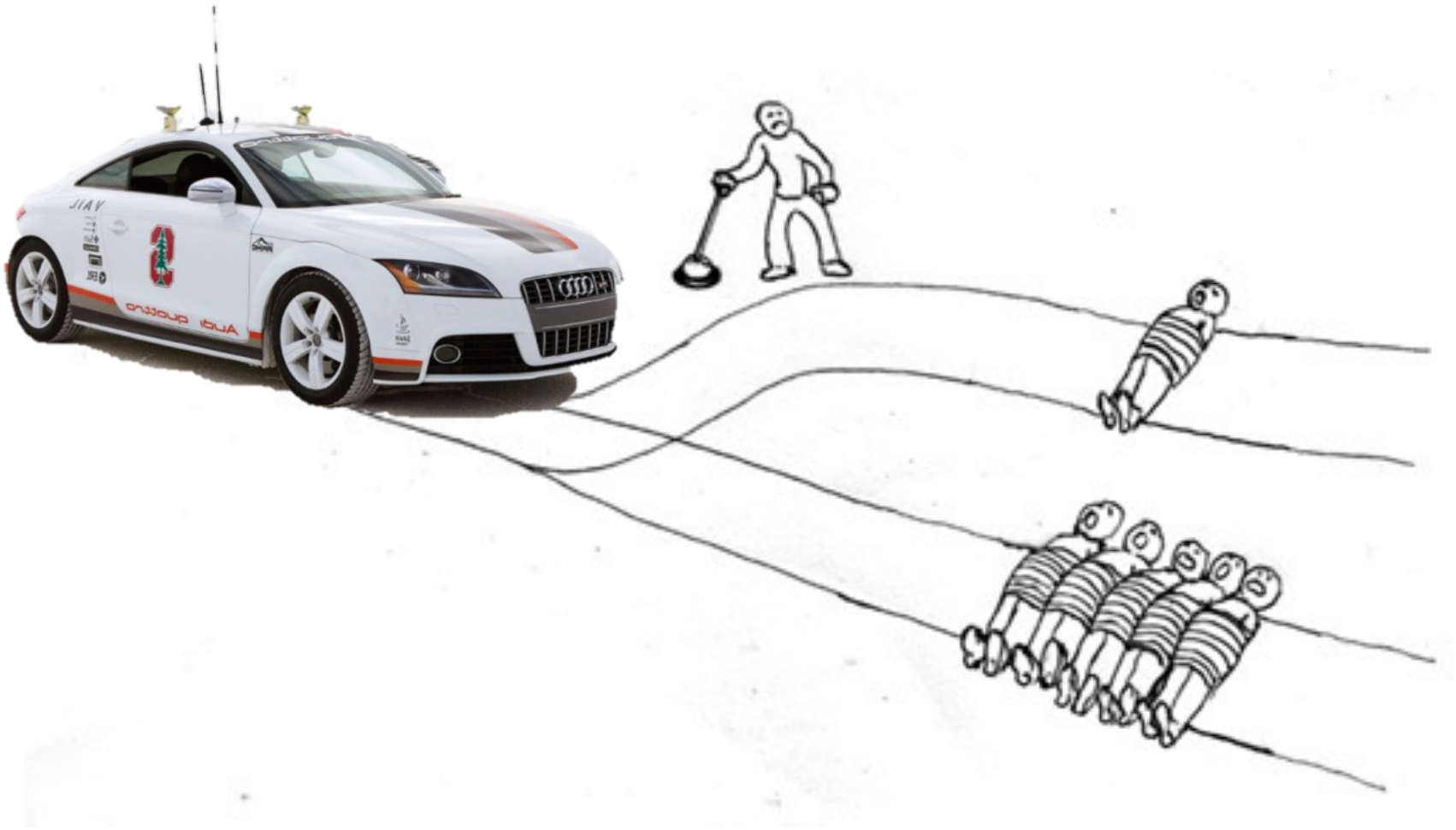


Probability: Most Desired Skill in Academia?

Most CS PhD students list their highest desiderata upon graduation as:

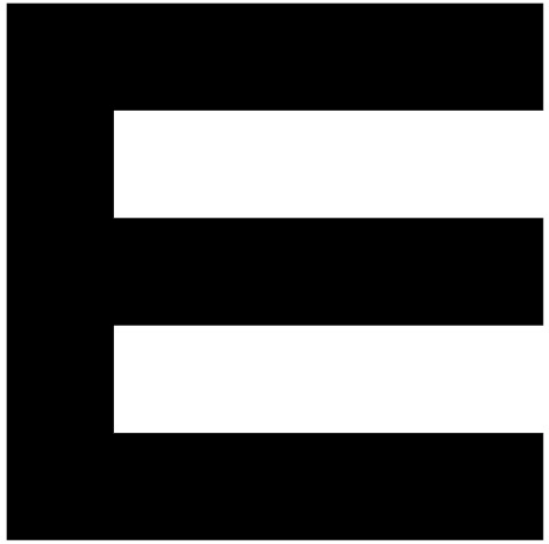
"Better understanding of probability"

Philosophy & Ethics of Modern AI Needs You



Piech & Cain, CS109, Stanford University

Solving Real Problems in CS109



Patient sees a series of letters of different font size, and for each, answers correct or incorrect

You decide that the vision tests given by eye doctors could have more precise results if we used an approach inspired by logistic regression. In a vision test a user looks at a letter with a particular font size and either correctly guesses the letter or incorrectly guesses the letter.

You assume that the probability that a particular patient is able to guess a letter correctly is:

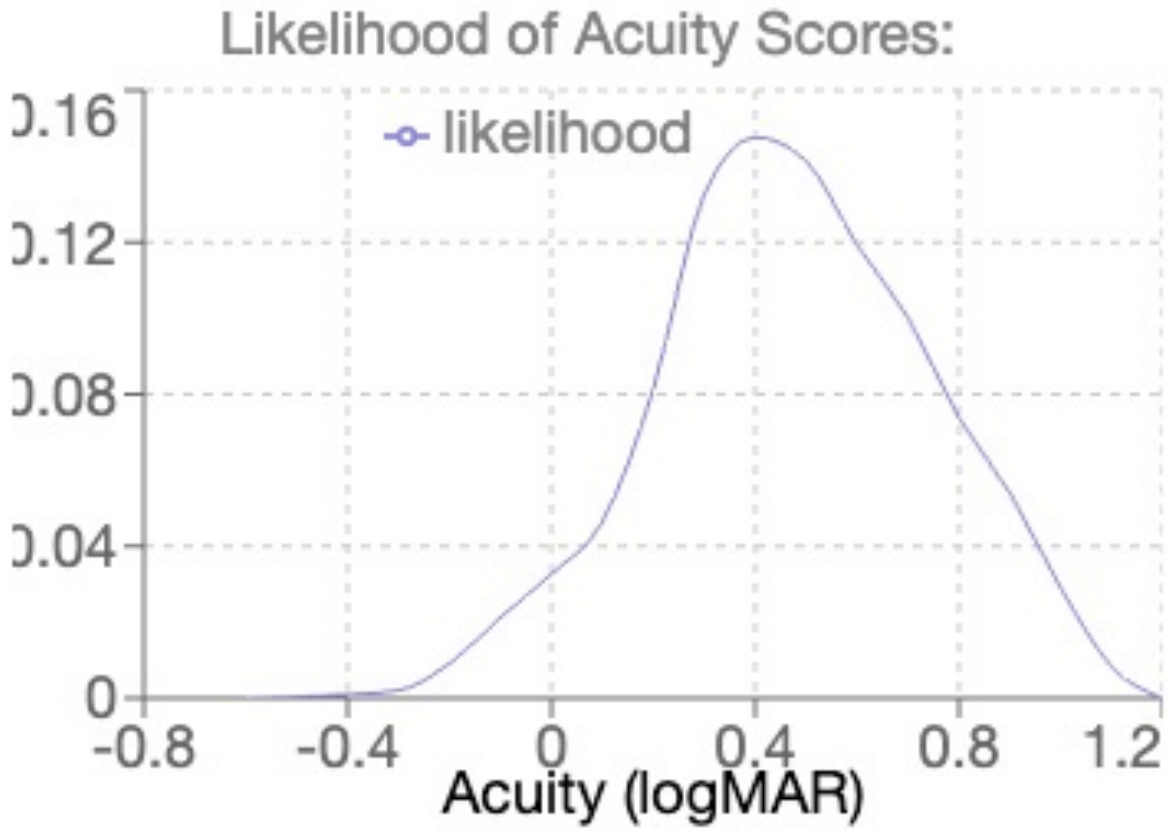
$$p = \sigma(\theta - f)$$

Where θ is the user's vision score and f is the font size of the letter.

Explain how you could estimate a user's vision score (θ) based on their 20 responses $(f^{(1)}, y^{(1)}) \ldots (f^{(20)}, y^{(20)})$, where $y^{(i)}$ is an indicator variable for whether the user correctly identified the ith letter and $f^{(i)}$ is the font size of the ith letter. Solve for any and all partial derivatives required by your answer.

Solving Real Problems in CS109

A patient has answered 20 “letter sizes” and got a few correct. What is your belief in how well they can see?



Solving Real Problems in CS109

The Stanford Acuity Test: A Precise Vision Test Using Bayesian Techniques and a Discovery in Human Visual Response

Chris Piech,[*1] **Ali Malik,**[*1] **Laura M Scott, Robert T Chang,**[2] **Charles Lin**[2]
[1]Department of Computer Science, Stanford University
[2] Department of Ophthalmology, Stanford University
{piech, malikali}@cs.stanford.edu, {rchang3, lincc}@stanford.edu

Abstract

Chart-based visual acuity measurements are used by billions of people to diagnose and guide treatment of vision impairment. However, the ubiquitous eye exam has no mechanism for reasoning about uncertainty and as such, suffers from a well-documented reproducibility problem. In this paper we make two core contributions. First, we uncover a new parametric probabilistic model of visual acuity response based on detailed measurements of patients with eye disease. Then, we present an adaptive, digital eye exam using modern artificial intelligence techniques which substantially reduces acuity exam error over existing approaches, while also introducing the novel ability to model its own uncertainty and incorporate prior beliefs. Using standard evaluation metrics, we estimate a 74% reduction in prediction error compared to the ubiquitous chart-based eye exam and up to 67% reduction compared to the previous best digital exam. For patients with eye disease, the novel ability to finely measure acuity from home could be a crucial part in early diagnosis. We provide a web implementation of our algorithm for anyone in the world to use. The insights in this paper also provide interesting implications for the field of psychometric Item Response Theory.

1 Introduction

Reliably measuring a person's visual ability is an essential component in the detection and treatment of eye diseases around the world. However, quantifying how well an individual can distinguish visual information is a surprisingly difficult task—without invasive techniques, physicians rely on chart-based eye exams where patients are asked visual questions and their responses observed.

Historically, vision has been evaluated by measuring a patient's *visual acuity*: a measure of the font size at which a patient can correctly identify letters shown a fixed distance away. Snellen, this statistic by asking the patie dentify the size ters correct. Thi

treatment of patients; yet, it suffers from some notable shortcomings. Acuity exams such as these exhibit high variance in their results due to the large role that chance plays in the final diagnosis, and the approximation error incurred by the need to discretise letter sizes on a chart. On the other hand, digital exams can show letter of any size and can *adaptively* make decisions based on intelligent probabilistic models. As such they have potential to address the shortcomings of analog charts.

While promising, contemporary digital exams have yet to dramatically improve accuracy over traditional chart-based approaches. The current best digital exam uses a psychometric Item Response Theory (IRT) algorithm for both selecting the next letter size to query and for making a final prediction of acuity. Under simulation analysis, this digital exam results in a 19% reduction in error over traditional chart-based approaches. The separate fields of reinforcement learning and psychometric IRT have independently explored how to effectively make decisions under uncertainty. By merging the good ideas from both disciplines we can develop a much better visual acuity test.

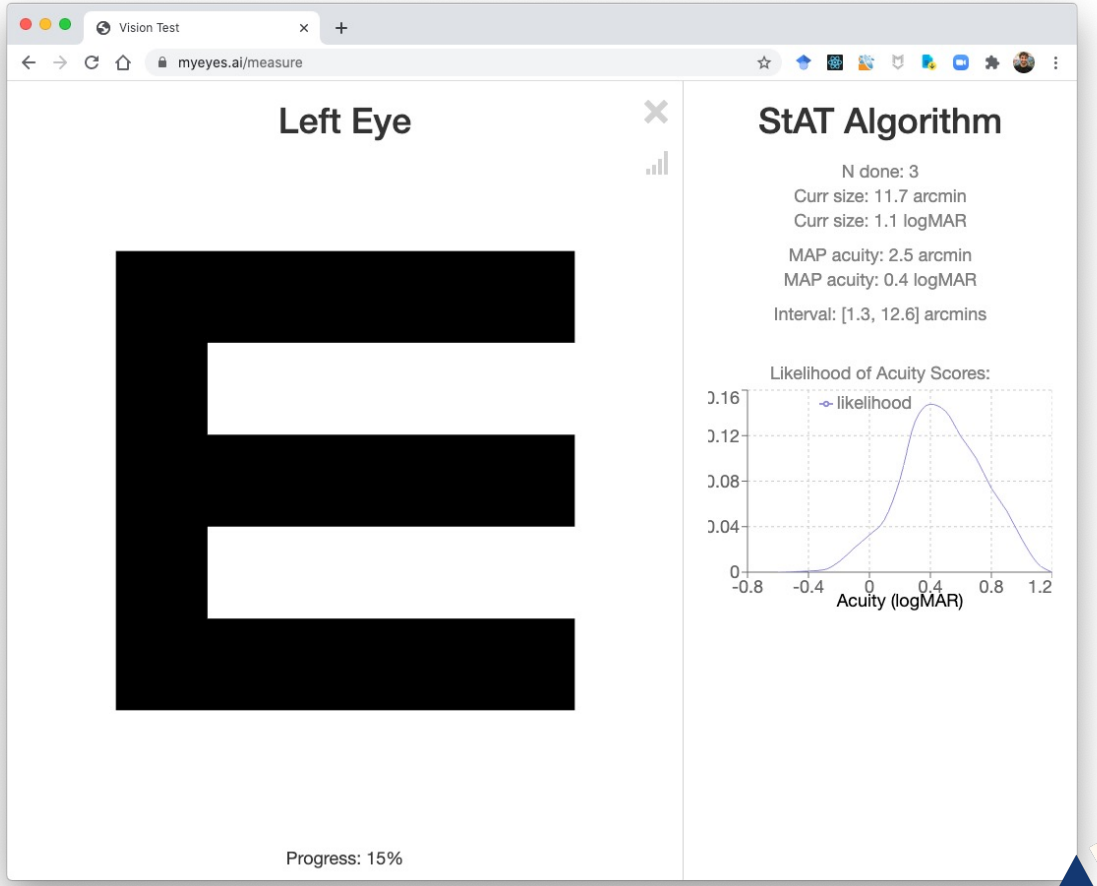
In this paper we make two main contributions. First, we revisit the human Visual Response Function—a function relating the size of a letter to the probability of a person identifying it correctly—and discover that it follows an interpretable parametric form that fits real patient data. Second, we present an algorithm to measure a person's acuity which uses several Bayesian techniques common in modern artificial intelligence. The algorithm, called the Stanford Acuity Test (StACT)[1], has the following novel features:

1. Uses the new parametric form of the human Visual Response Function.
2. Returns a soft inference prediction of the patient's acuity, confidence in the final

ıg algorithm to adap-
wn to a user. This ef-
acuity belief.

CT, was named after
ed. We continue in this

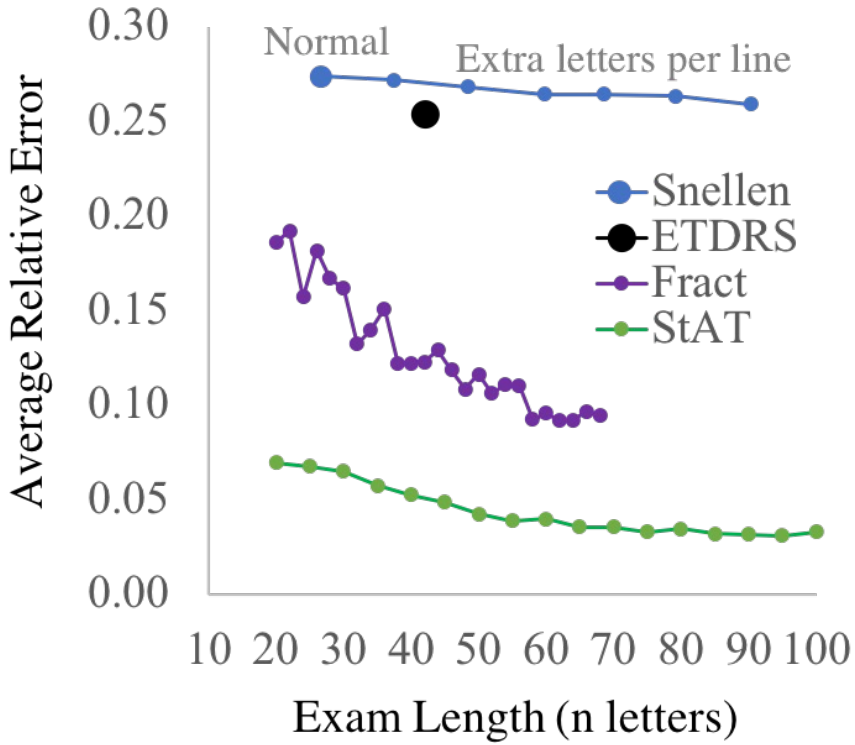
*Equal contrit
Copyright © 202
Intelligence (ww

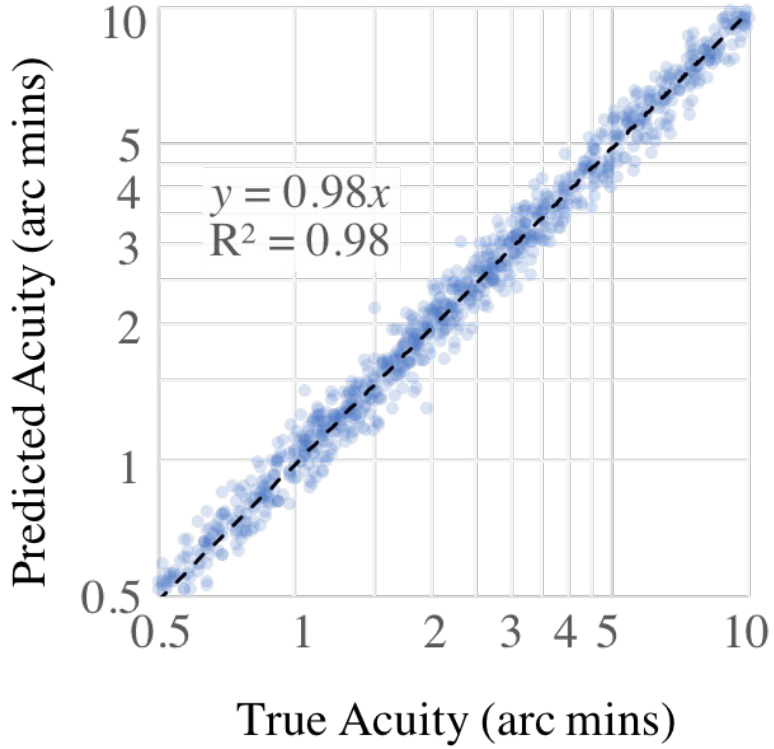


Science

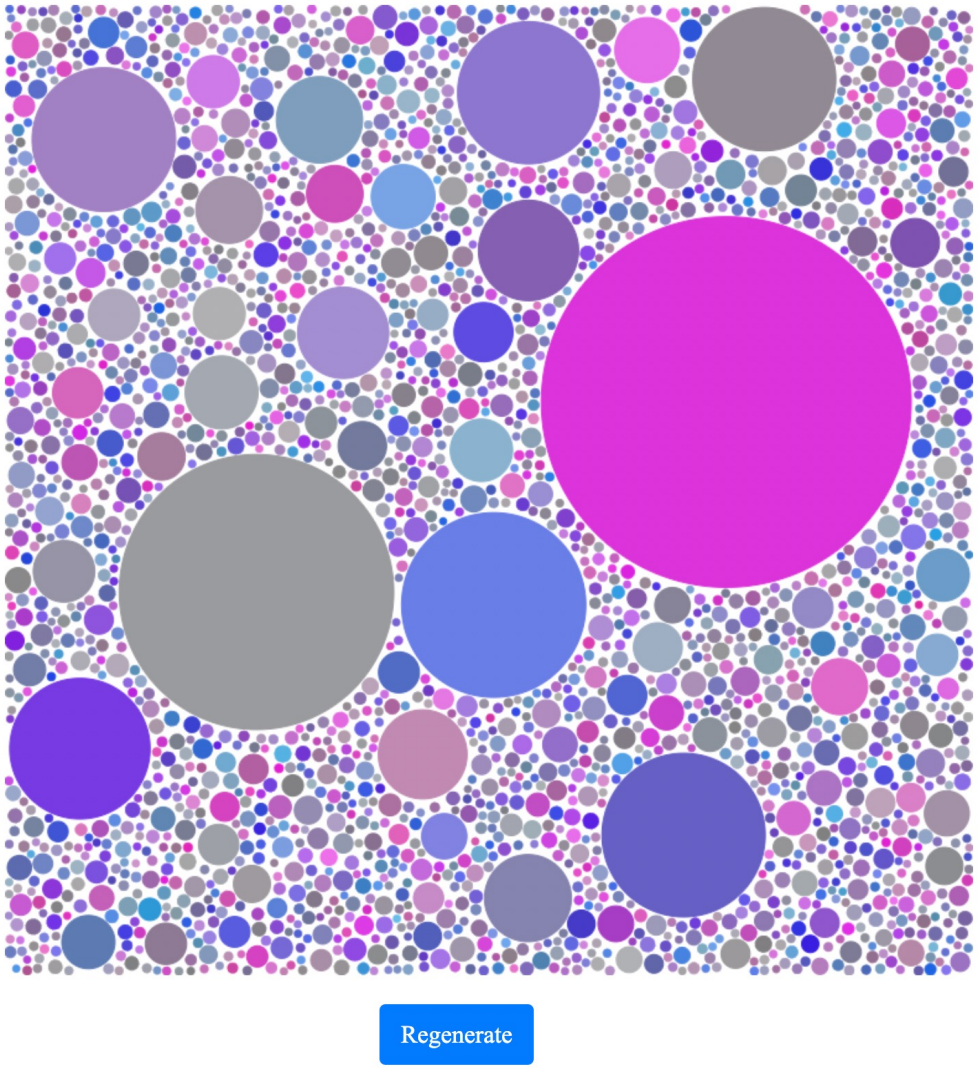
THE LANCET

Now state of the art for eye exam theory





What is on a typical exam?



1. Algorithmic Art
2. Lucky Events
3. Supply Chain Decision Making
4. P-Hacking
5. Chess.com Puzzle Ability
6. ML Calibration

https://chrispiech.github.io/probabilityForComputerScientists/en/examples/algorithmic_art/

Probability is particularly fun
because it is often not intuitive

Intuition and Probability: Frenemies

A patient has a
positive Zika test.

What is the probability they have zika?

- *0.8% of people have zika*
- *Test has 90% positive rate for people with zika*
- *Test has 7% positive rate for people without zika*

The right answer is 9%

CS109 View of Probability

Teach you how to write programs
to solve probability problems
that most people are not able to write.

AND

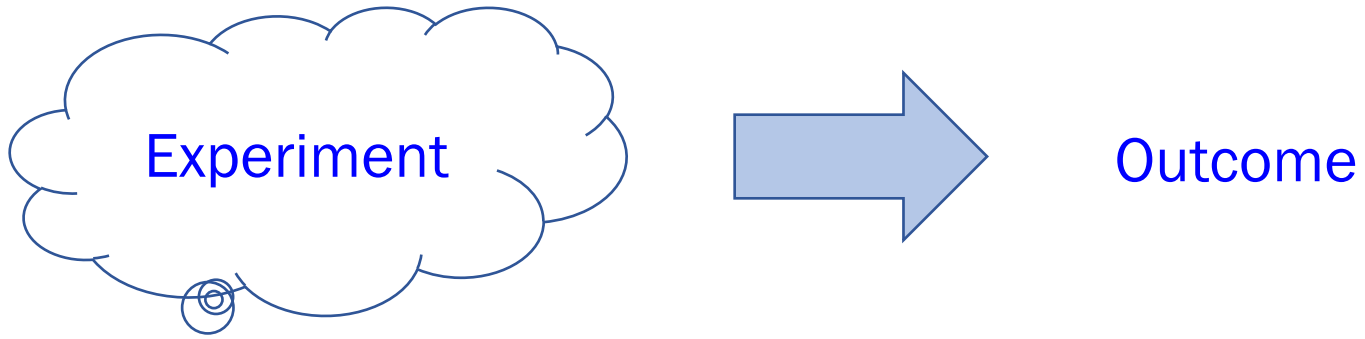
Teach you the theory you need to do the math
that most people are not able to do.


Probabilistic Models
Uncertainty Theory
Machine Learning
Random Variables
Core Probability
Counting
Let the journey begin…

Counting

Goal: Count Outcomes of Experiments

- **Experiment:** any situation where what's going to happen is uncertain
- **Outcome:** one possible thing that could happen



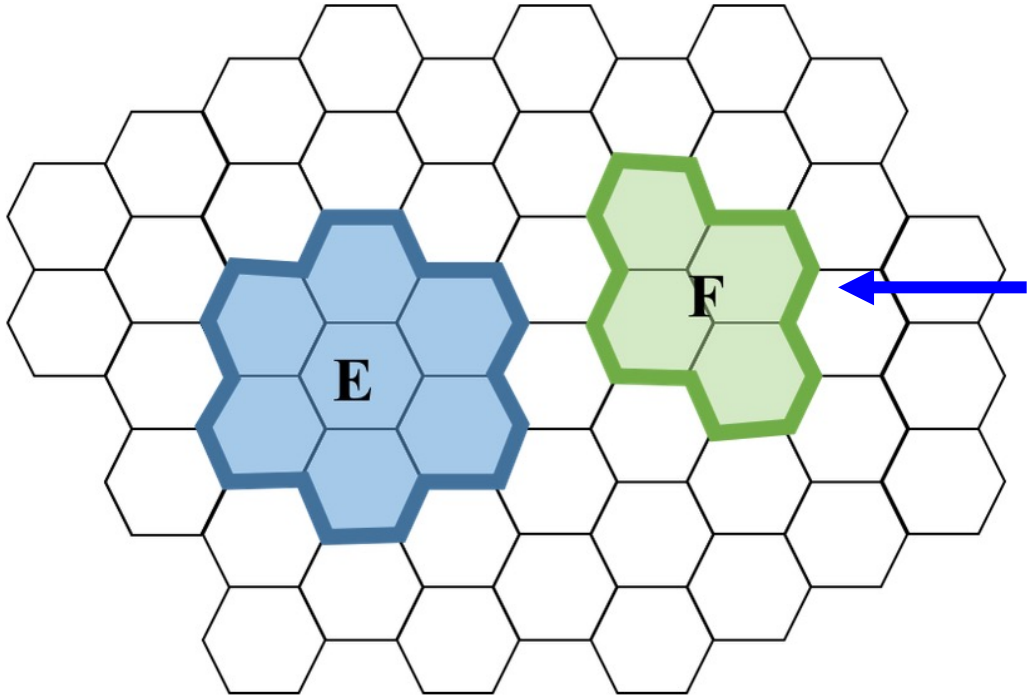
Experiments have *sets* of outcomes, containing groups of things that could possibly happen.

Step 1 of doing probability: counting the number of possible outcomes.

Events: Interesting Subsets of Outcomes

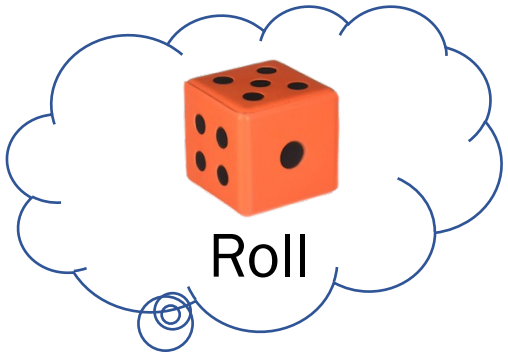
- **Event:** some subset of all possible outcomes that we care about

This is the entire **sample space**: all possible outcomes

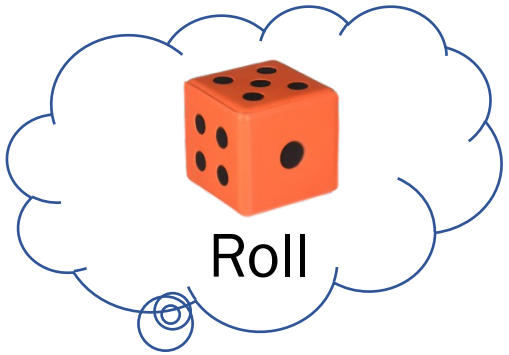


Here is one event

Example Outcomes



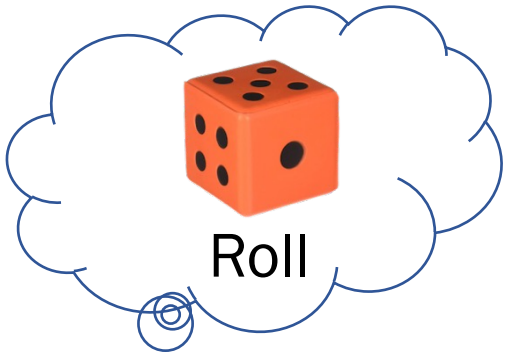
Example Outcomes



6

{1, 2, 3,
4, 5, 6}

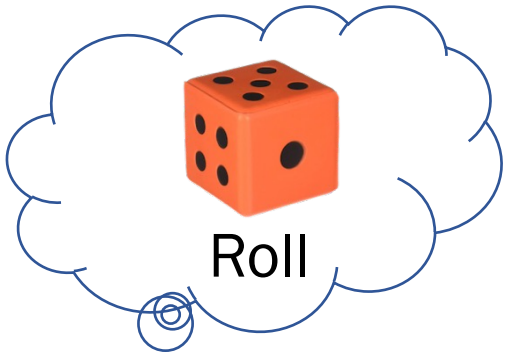
Example Outcomes



6

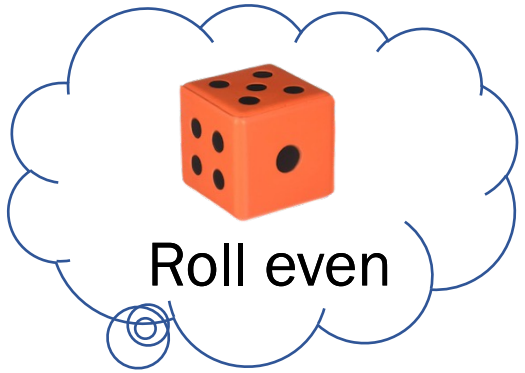
{1, 2, 3,
4, 5, 6}

Example Outcomes



6

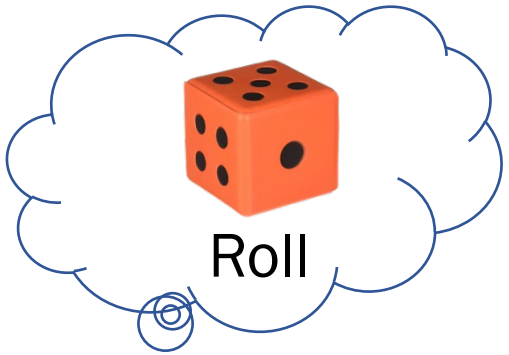
{1, 2, 3, 4, 5, 6}



3

{2,4,6}

Example Outcomes

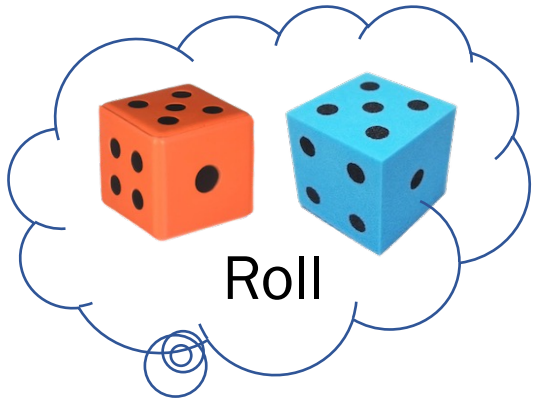


6

{1, 2, 3, 4, 5, 6}

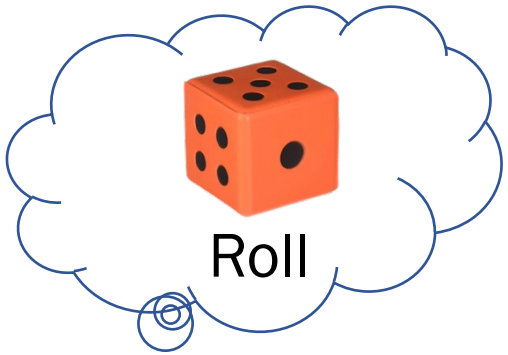
3

{2,4,6}



Tip #1: start by imagining one outcome

Example Outcomes

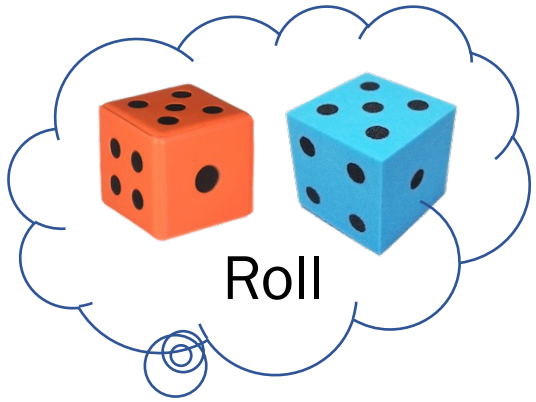


6

{1, 2, 3, 4, 5, 6}

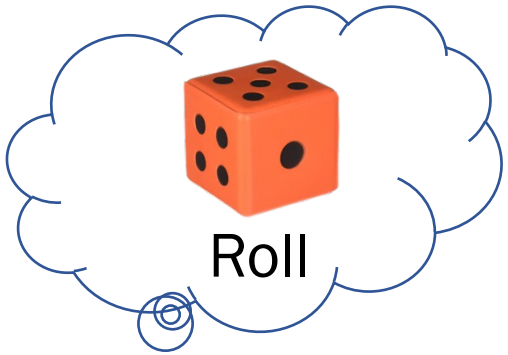
3

{2,4,6}



{(1, 1) , (1, 2), (1, 3), (1, 4), (1, 5), (1, 6),
(2, 1) , (2, 2), (2, 3), (2, 4), (2, 5), (2, 6),
(3, 1) , (3, 2), (3, 3), (3, 4), (3, 5), (3, 6),
(4, 1) , (4, 2), (4, 3), (4, 4), (4, 5), (4, 6),
(5, 1) , (5, 2), (5, 3), (5, 4), (5, 5), (5, 6),
(6, 1) , (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)}

Example Outcomes

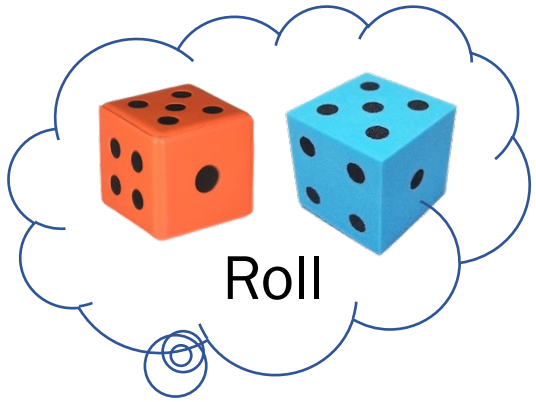


6

{1, 2, 3, 4, 5, 6}

3

{2,4,6}



36

{(1, 1) , (1, 2), (1, 3), (1, 4), (1, 5), (1, 6),
(2, 1) , (2, 2), (2, 3), (2, 4), (2, 5), (2, 6),
(3, 1) , (3, 2), (3, 3), (3, 4), (3, 5), (3, 6),
(4, 1) , (4, 2), (4, 3), (4, 4), (4, 5), (4, 6),
(5, 1) , (5, 2), (5, 3), (5, 4), (5, 5), (5, 6),
(6, 1) , (6, 2), (6, 3), (6, 4), (6, 5), (6, 6)}

Tip #2: Think about a generative story…

Step/Product Rule of Counting

Two-step experiment

- If an experiment has two steps, where
  - Step 1's outcomes make up Set A, where $|A| = m$, and Step 2's outcomes make up Set B, where $|B| = n$,
  - and $|B|$ is unaffected by the outcome of Step 1,

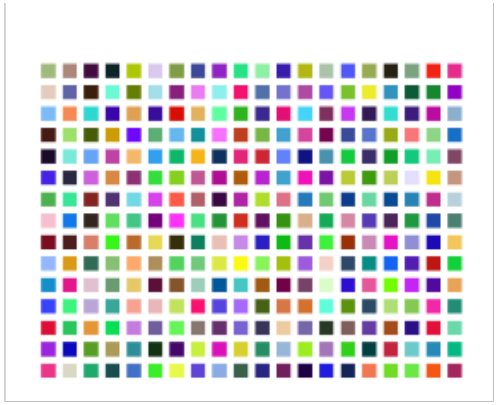
- Then the number of outcomes of the experiment is

$$|A||B| = mn.$$

How Many Unique Images?

Each pixel can be one of 17 million distinct colors

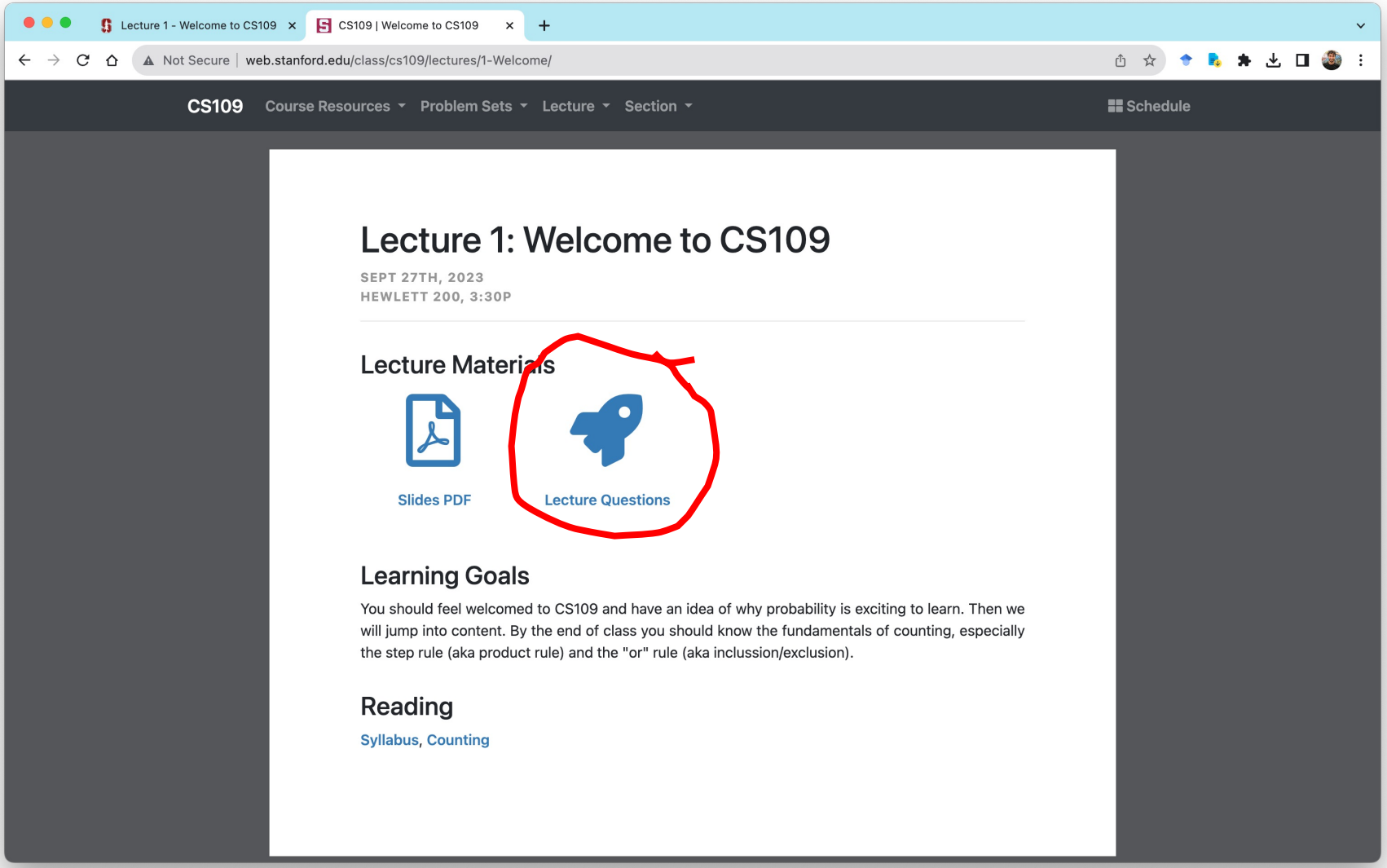
(a) 12 million pixels



(b) 300 pixels

(c) 12 pixels

The First Lecture Concept Check

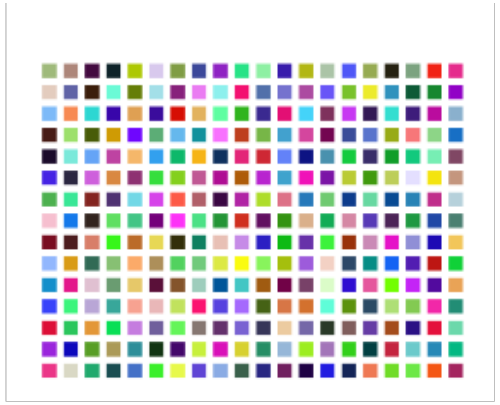


How Many Unique Images?

Each pixel can be one of 17 million distinct colors

(a) 12 million pixels

$$\approx 10^{86696638}$$



(b) 300 pixels

$$\approx 10^{2167}$$

(c) 12 pixels

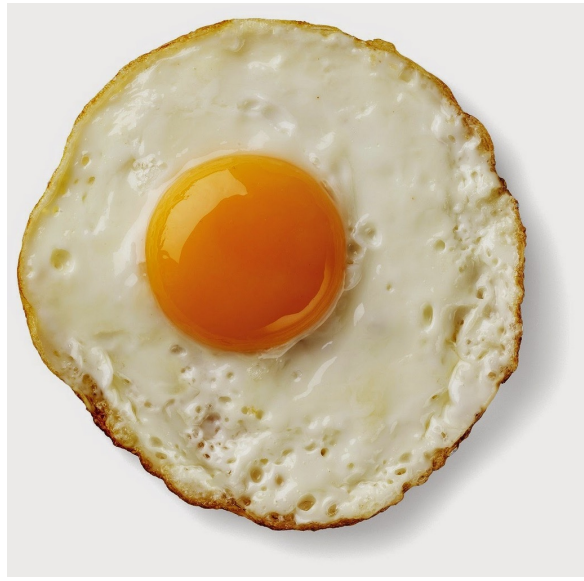
$$\approx 10^{86}$$

$$(17 \text{ million})^n$$

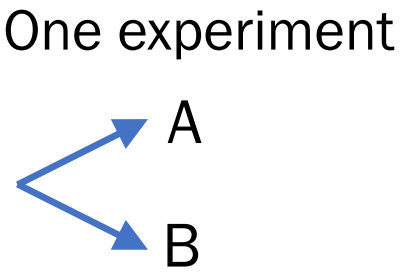

10^{80}

How Many Ways To Cook Eggs?

Question: Eggs can be boiled, fried, scrambled, or poached. There are 3 ways to boil eggs, 4 ways to fry eggs, 2 ways to scramble eggs, and one way to poach eggs. How many total egg-cooking options are there?



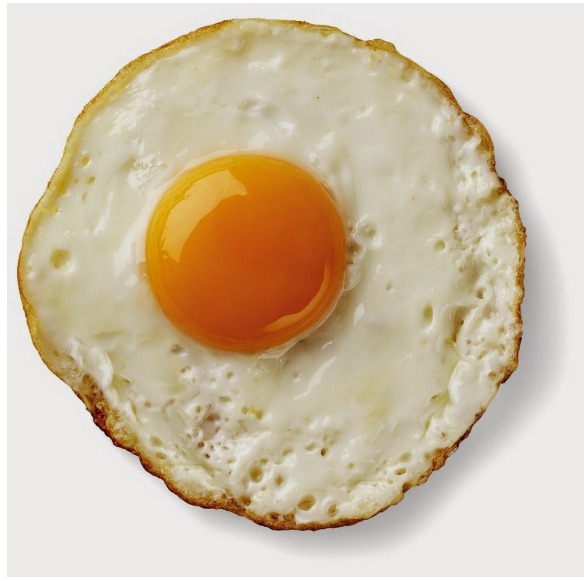
The "Or" Rule, Part 1



- If the outcome of an experiment can be either from
  - Set A, where $|A| = m$,
  - **or** Set B, where $|B| = n$,
  - where $A \cap B = \emptyset$ (no overlap)
- Then the number of outcomes of the experiment is
  - $|A| + |B| = m + n$.

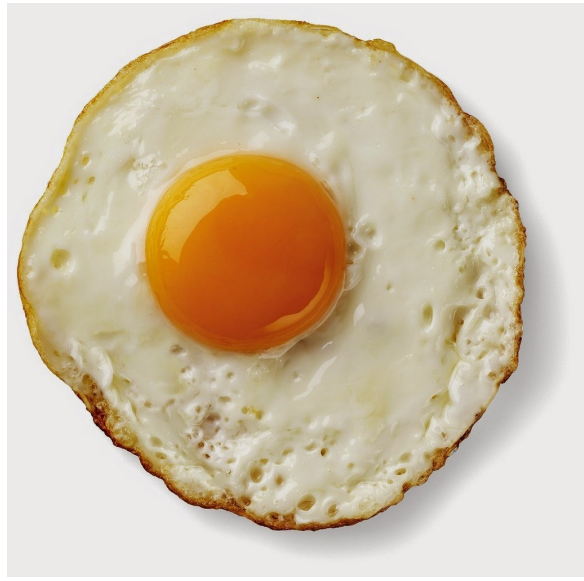
How Many Ways To Cook Eggs?

Question: Eggs can be boiled, fried, scrambled, or poached. There are 3 ways to boil eggs, 4 ways to fry eggs, 2 ways to scramble eggs, and one way to poach eggs. How many total egg-cooking options are there?



How Many Ways To Cook Eggs?

Question: Eggs can be boiled, fried, scrambled, or poached. There are 3 ways to boil eggs, 4 ways to fry eggs, 2 ways to scramble eggs, and one way to poach eggs. How many total egg-cooking options are there?



Answer: 3 + 4 + 2 + 1

How Many Bit Strings?

Problem: A 6-bit string (made of 1s and 0s) is sent over a network. The valid set of strings recognized by the receiver must either start with "01" or end with "10". How many such strings are there?

How Many Bit Strings?

Problem: A 6-bit string (made of 1s and 0s) is sent over a network. The valid set of strings recognized by the receiver must either start with "01" or end with "10". How many such strings are there?

Set A

```
010000
010001
010010
010011
010100
010101
010110
010111
011000
011001
011010
011011
011100
011101
011110
011111
```

Set B

```
000010
000110
001010
001110
010010
010110
011010
011110
100010
100110
101010
101110
110010
110110
111010
111110
```

How Many Bit Strings?

Problem: A 6-bit string (made of 1s and 0s) is sent over a network. The valid set of strings recognized by the receiver must either start with "01" or end with "10". How many such strings are there?

2^4 start with 01

```
010000
010001
010010
010011
010100
010101
010110
010111
011000
011001
011010
011011
011100
011101
011110
011111
```

Set A

2^4 end with 10

```
000010
000110
001010
001110
010010
010110
011010
011110
100010
100110
101010
101110
110010
110110
111010
111110
```

Set B

How Many Bit Strings?

Problem: A 6-bit string (made of 1s and 0s) is sent over a network. The valid set of strings recognized by the receiver must either start with "01" or end with "10". How many such strings are there?

2^4 start with 01

010000
010001
010010
010011
010100
010101
010110
010111
011000
011001
011010
011011
011100
011101
011110
011111

Set A

2^4 end with 10

000010
000110
001010
001110
010010
010110
011010
011110
100010
100110
101010
101110
110010
110110
111010
111110

Set B

How Many Bit Strings?

Problem: A 6-bit string (made of 1s and 0s) is sent over a network. The valid set of strings recognized by the receiver must either start with "01" or end with "10". How many such strings are there?

Answer

$$N = |A| + |B| - |A \text{ and } B|$$
$$= 16 + 16 - 4$$
$$= 28$$

2^4 start with 01

010000
010001
010010
010011
010100
010101
010110
010111
011000
011001
011010
011011
011100
011101
011110
011111

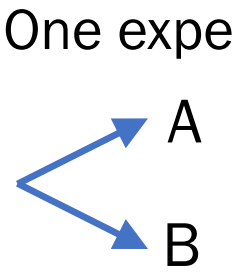
Set A

2^4 end with 10

000010
000110
001010
001110
010010
010110
011010
011110
100010
100110
101010
101110
110010
110110
111010
111110

Set B

The "Or" Rule (aka Inclusion/Exclusion)



- If the outcome of an experiment can be either from
  - Set A, where $|A| = m$,
  - **or** Set B, where $|B| = n$,
  - where $A \cap B$ $might\ not\ be\ empty$,
- Then the number of outcomes of the experiment is
  - N = $|A| + |B| - |A \cap B|$.

The Core Counting Rules

Counting with steps

Definition: Step Rule of Counting (aka Product Rule of Counting)

If an experiment has two parts, where the first part can result in one of m outcomes and the second part can result in one of n outcomes regardless of the outcome of the first part, then the total number of outcomes for the experiment is $m \cdot n$.

Counting with "or"

Definition: Inclusion Exclusion Counting

If the outcome of an experiment can either be drawn from set A or set B, and sets A and B may potentially overlap (i.e., it is not the case that A and B are mutually exclusive), then the number of outcomes of the experiment is $|A \text{ or } B| = |A| + |B| - |A \text{ and } B|$.

Challenge Problem

How many *different* orderings of letters are possible for the string BOBA?

BOBA, ABOB, OBBA…

Have fun pondering!