Introduction

Ling 282/482: Deep Learning for Computational Linguistics

C.M. Downey

Fall 2025

Overview

Overview

- Overall theme: **Neural Language Models from the Ground Up**

Overview

- Overall theme: **Neural Language Models from the Ground Up**
- Breaking this down:

Overview

- Overall theme: **Neural Language Models from the Ground Up**
- Breaking this down:
  - Neural ≈ "based on **Deep Learning**" (I'll use these almost interchangeably)

Overview

- Overall theme: **Neural Language Models from the Ground Up**
- Breaking this down:
  - Neural ≈ "based on **Deep Learning**" (I'll use these almost interchangeably)
  - Language Models: **probabilistic** models of a **sequence of symbols**

Overview

- Overall theme: **Neural Language Models from the Ground Up**
- Breaking this down:
  - Neural ≈ "based on **Deep Learning**" (I'll use these almost interchangeably)
  - Language Models: **probabilistic** models of a **sequence of symbols**
  - "from the ground up": starting with **mathematical foundations** and working up to **modern LLMs/chatbots**

Overview

- Overall theme: **Neural Language Models from the Ground Up**
- Breaking this down:
  - Neural ≈ "based on **Deep Learning**" (I'll use these almost interchangeably)
  - Language Models: **probabilistic** models of a **sequence of symbols**
  - "from the ground up": starting with **mathematical foundations** and working up to **modern LLMs/chatbots**
- This is meant to be a **foundational introduction** to the area

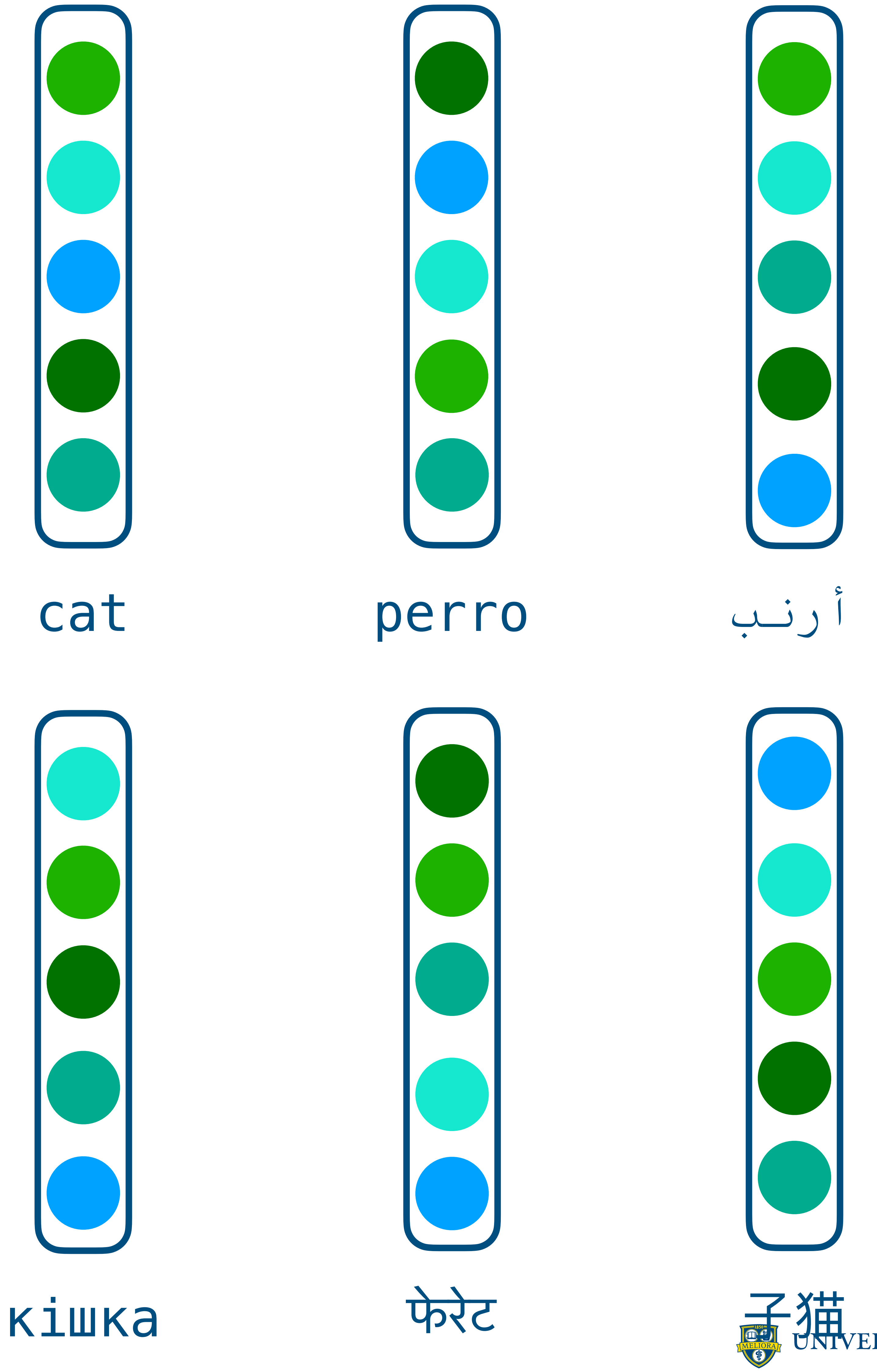
Overview

- Overall theme: **Neural Language Models from the Ground Up**
- Breaking this down:
  - Neural ≈ "based on **Deep Learning**" (I'll use these almost interchangeably)
  - Language Models: **probabilistic** models of a **sequence of symbols**
  - "from the ground up": starting with **mathematical foundations** and working up to **modern LLMs/chatbots**
- This is meant to be a **foundational introduction** to the area
  - **Assumed knowledge**: derivatives, basic probability, Python, git/github

Neural Networks / Deep Learning

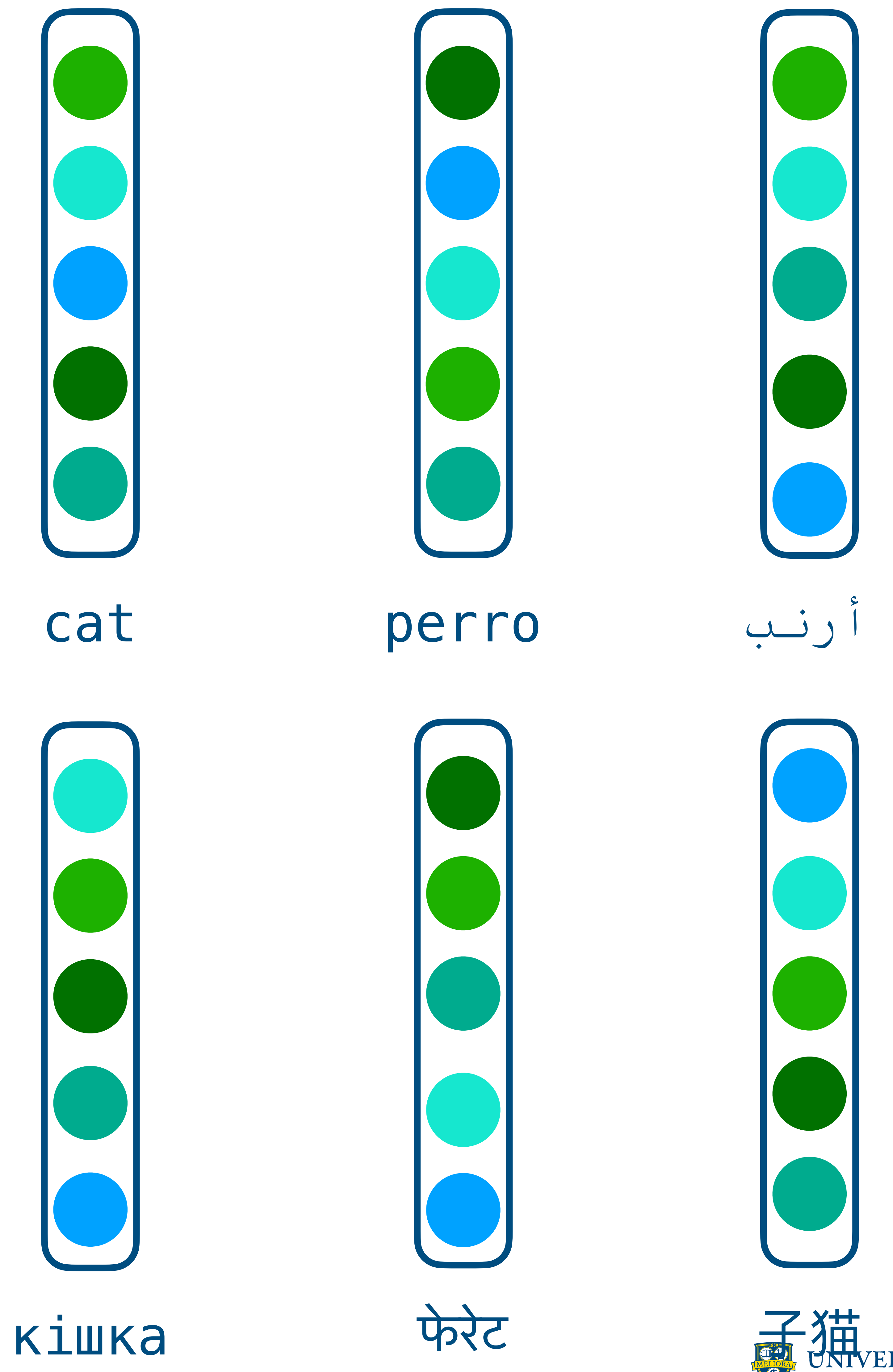
Neural Networks / Deep Learning

- Neural Networks operate on **vectors**
  - **lists of numbers** (more next time)



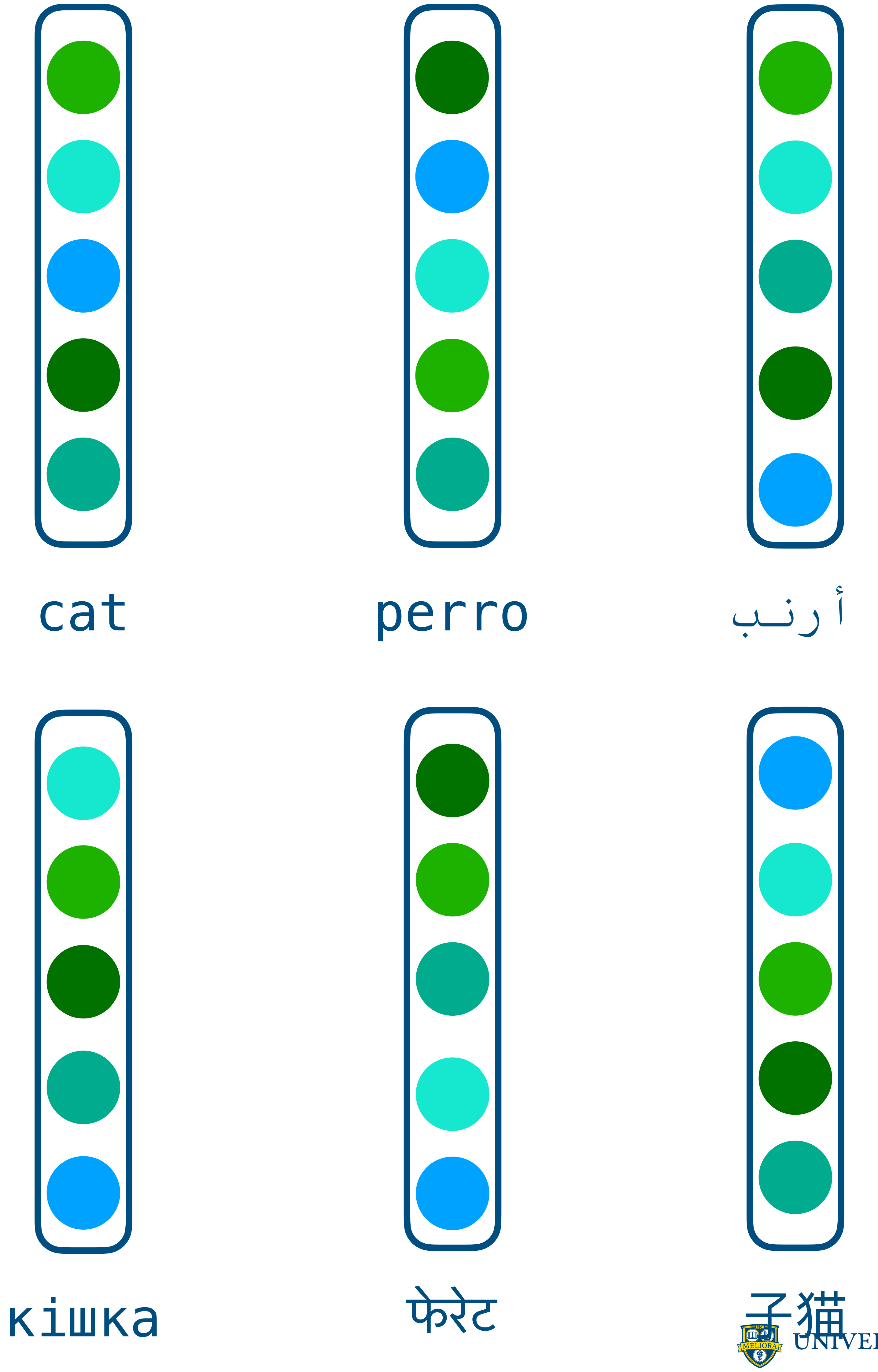
Neural Networks / Deep Learning

- Neural Networks operate on **vectors**
  - **lists of numbers** (more next time)
- Vectors can represent **words, concepts, entities, decisions,** etc.



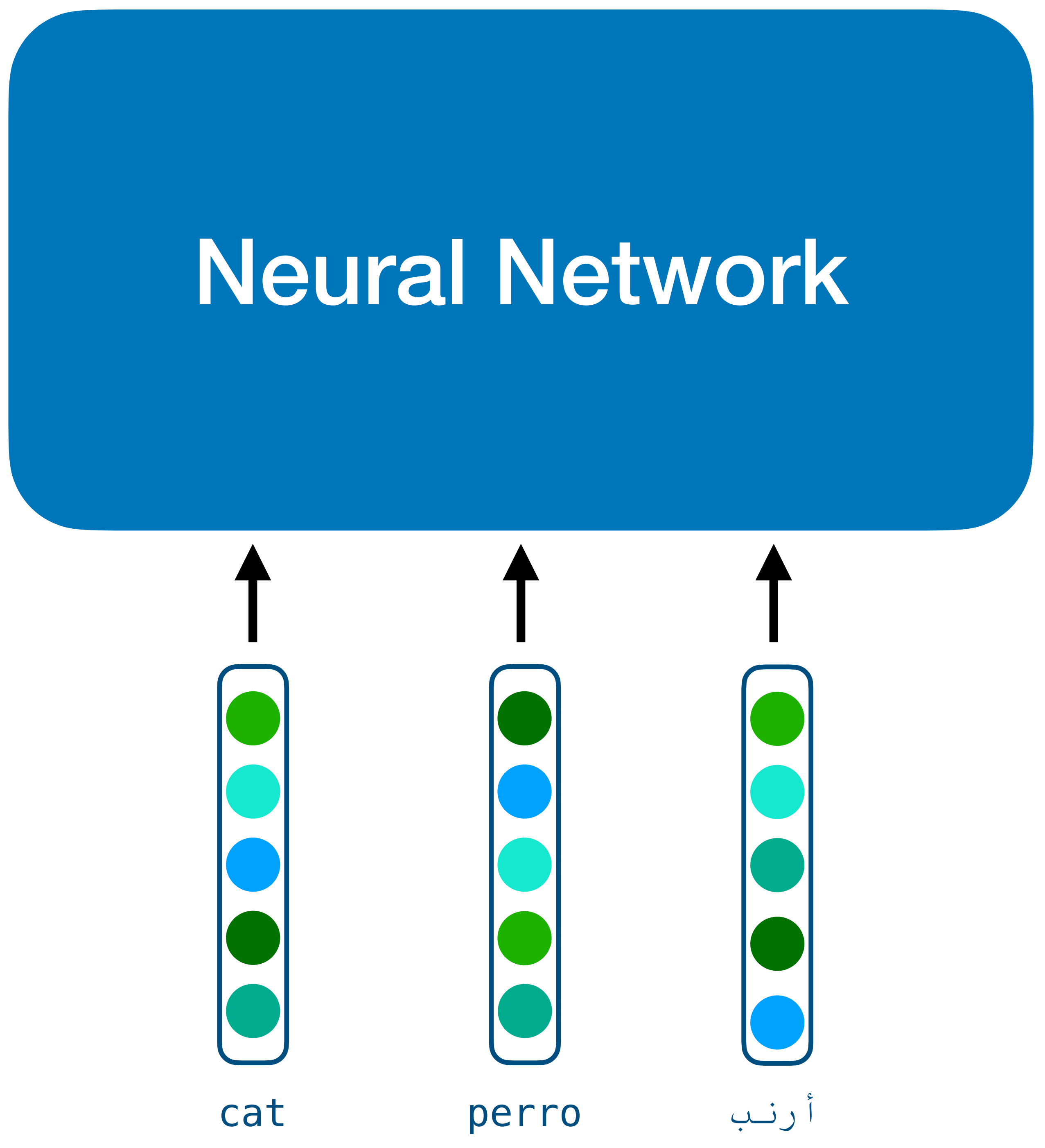
Neural Networks / Deep Learning

- Neural Networks operate on **vectors**
  - **lists of numbers** (more next time)
- Vectors can represent **words, concepts, entities, decisions,** etc.
- NNs take in vectors and **transform** them into **new vectors**



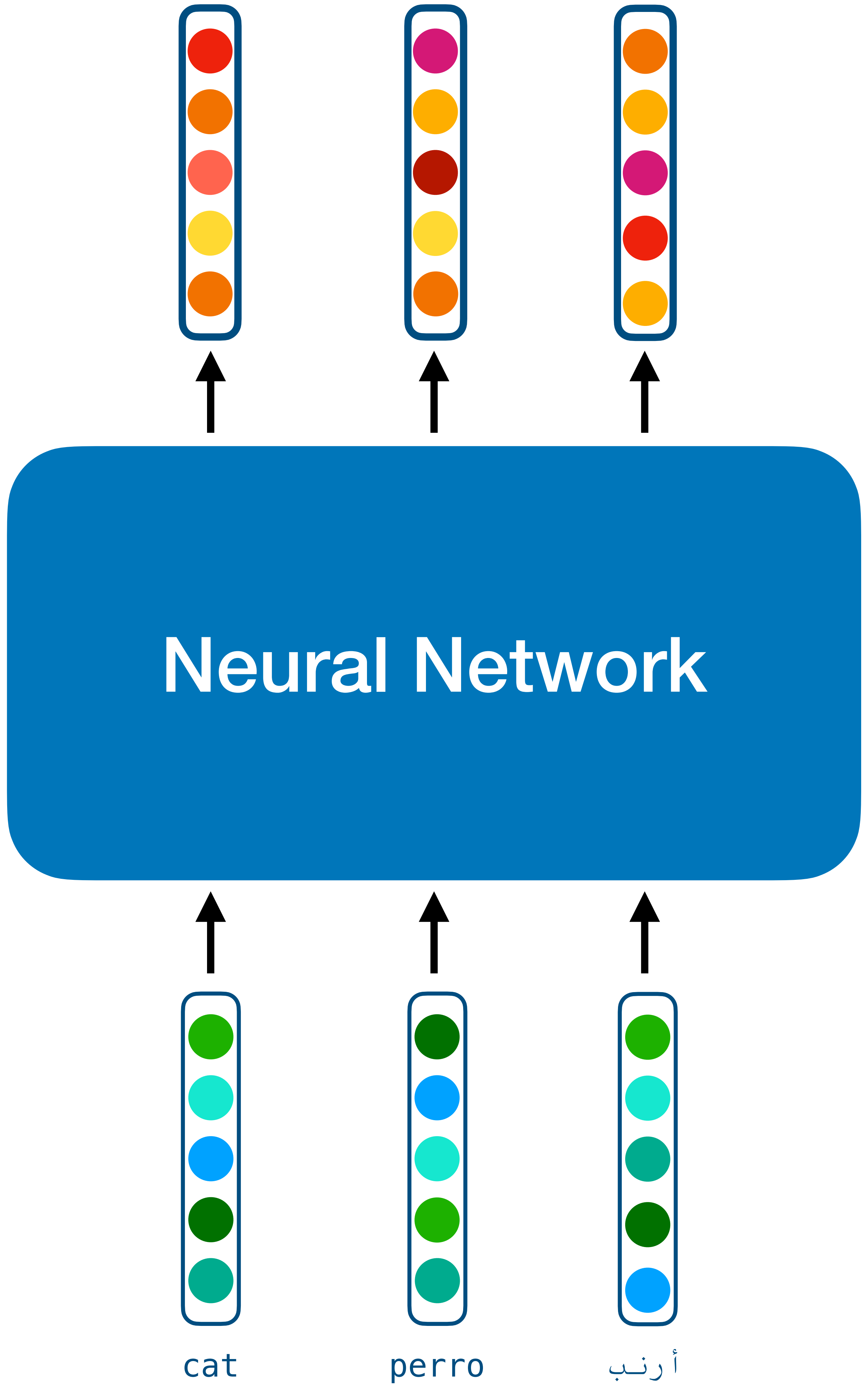
Neural Networks / Deep Learning

- Neural Networks operate on **vectors**
  - **lists of numbers** (more next time)
- Vectors can represent **words, concepts, entities, decisions,** etc.
- NNs take in vectors and **transform** them into **new vectors**



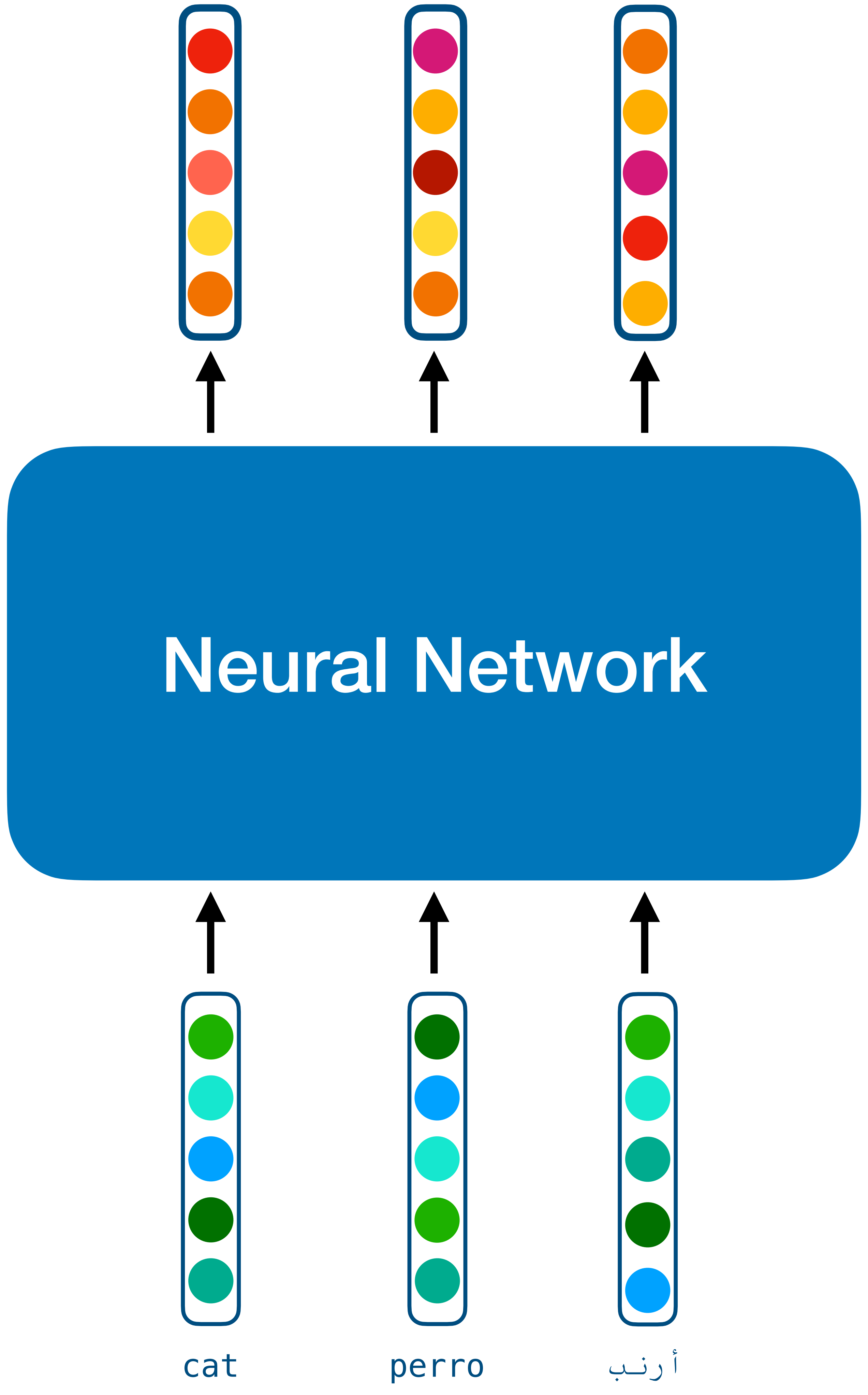
Neural Networks / Deep Learning

- Neural Networks operate on **vectors**
  - **lists of numbers** (more next time)
- Vectors can represent **words, concepts, entities, decisions,** etc.
- NNs take in vectors and **transform** them into **new vectors**



Neural Networks / Deep Learning

- Neural Networks operate on **vectors**
  - **lists of numbers** (more next time)
- Vectors can represent **words, concepts, entities, decisions,** etc.
- NNs take in vectors and **transform** them into **new vectors**
- Using NNs with **multiple layers** is referred to as **Deep Learning**



What is Language Modeling?

What word comes next?

The ____

What is Language Modeling?

What word comes next?

The ____

class
woman
axon
green
colorless
great
⋮

What is Language Modeling?

What word comes next?

The _____

Nouns

$\boxed{\begin{array}{c}\text{class}\\ \text{woman}\\ \text{axon}\end{array}}$

green
colorless
great
⋮

What is Language Modeling?

What word comes next?

The _____

Nouns

class
woman
axon

Adjectives

green
colorless
great

⋮

What is Language Modeling?

What word comes next?

The _____

Nouns: class, woman, axon

Adjectives: green, colorless, great

⋮

We can predict which **Parts of Speech** are likely!

What is Language Modeling?

What word comes next?

The _____

class
woman
axon
green
colorless
great
⋮

What is Language Modeling?

What word comes next?

The calico ____

What is Language Modeling?

What word comes next?

The calico ____

cat
coat
fur
hair
beans
critter
⋮

What is Language Modeling?

What word comes next?

The calico ____

cat
coat
fur
hair
beans
critter
⋮

Sometimes a **single word** will be almost certain

What is Language Modeling?

What word comes next?

The calico ____

cat
coat
fur
hair
beans
critter
⋮

What is Language Modeling?

What word comes next?

The calico cat ____

What is Language Modeling?

What word comes next?

The calico cat ____

is
was
has
ran
sat
does
⋮

What is Language Modeling?

What word comes next?

The calico cat _____

is
was
has
ran
sat
does
⋮

Verbs

What is Language Modeling?

What word comes next?

The calico cat ____

is
was
has
ran
sat
does
⋮

What is Language Modeling?

What word comes next?

The calico cat sits _____

What is Language Modeling?

What word comes next?

The calico cat sits ____

on
in
with
still
sat
does
⋮

What is Language Modeling?

What word comes next?

The calico cat sits _____

on
in
with
still
sat
does
⋮

Prepositions

What is Language Modeling?

What word comes next?

The calico cat sits _____

on
in
with
still
sat
does
⋮

What is Language Modeling?

What word comes next?

The calico cat sits on _____

What is Language Modeling?

What word comes next?

The calico cat sits on _____

a
the
my
her
me
its
⋮

What is Language Modeling?

What word comes next?

The calico cat sits on the ____

What is Language Modeling?

What word comes next?

The calico cat sits on the ____

chair
ledge
window
mat
high
soft
⋮

What is Language Modeling?

What word comes next?

The calico cat sits on the ____

chair
ledge
window
mat
high
soft
⋮

places a cat might like

What is Language Modeling?

What word comes next?

The calico cat sits on the ____

chair
ledge
window
mat
high
soft
⋮

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny _____

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny _____

window
patio
porch
spot
ledge
roof
⋮

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny _____

window
patio
porch
spot
ledge
roof
⋮

potentially sunny

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny _____

window
patio
porch
spot
ledge
roof
⋮

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny ledge _____

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny ledge ____

.
!
</s>
every
and
all
⋮

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny ledge lately

What is Language Modeling?

What word comes next?

The calico cat sits on the sunny ledge lately

This is Language Modeling

Why bother with Language Modeling?

Why bother with Language Modeling?

- To make better predictions, an LM encodes **grammatical, semantic, and "real-world" knowledge** at every step

Why bother with Language Modeling?

- To make better predictions, an LM encodes **grammatical, semantic, and "real-world" knowledge** at every step
  - We'll see this makes them great **general-purpose models**

Why bother with Language Modeling?

- To make better predictions, an LM encodes **grammatical, semantic, and "real-world" knowledge** at every step
  - We'll see this makes them great **general-purpose models**
  - Also **interesting to Linguists** for the types of language structure they encode

Why bother with Language Modeling?

- To make better predictions, an LM encodes **grammatical, semantic, and "real-world" knowledge** at every step
  - We'll see this makes them great **general-purpose models**
  - Also **interesting to Linguists** for the types of language structure they encode
  - *Might* be conceivable as **models of human competence** (though this is highly controversial)

Why bother with Language Modeling?

- To make better predictions, an LM encodes **grammatical, semantic, and "real-world" knowledge** at every step
  - We'll see this makes them great **general-purpose models**
  - Also **interesting to Linguists** for the types of language structure they encode
  - *Might* be conceivable as **models of human competence** (though this is highly controversial)
- Making good predictions allows them to **generate new language**

Why bother with Language Modeling?

- To make better predictions, an LM encodes **grammatical, semantic, and "real-world" knowledge** at every step
  - We'll see this makes them great **general-purpose models**
  - Also **interesting to Linguists** for the types of language structure they encode
  - *Might* be conceivable as **models of human competence** (though this is highly controversial)
- Making good predictions allows them to **generate new language**
  - This has caused the explosion of **Generative AI** and **Chatbots**

Language Modeling (Technical Definition)

Language Modeling (Technical Definition)

- A Language Model makes a **probabilistic prediction** about a **missing component** of a **sequence of symbols**

Language Modeling (Technical Definition)

- A Language Model makes a **probabilistic prediction** about a **missing component** of a **sequence of symbols**
- "probabilistic": assigns a **probability** to **each possible prediction**

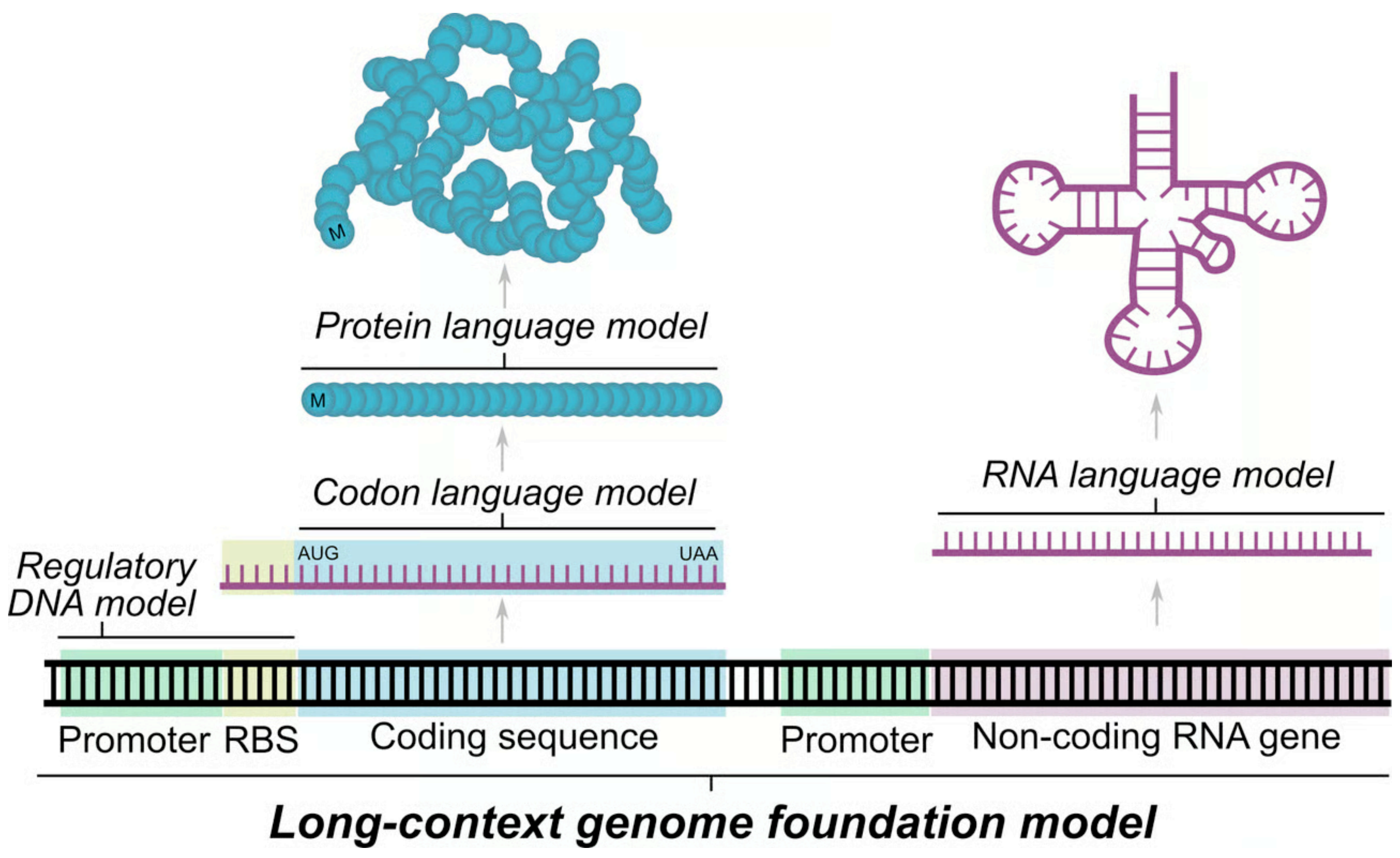
Language Modeling (Technical Definition)

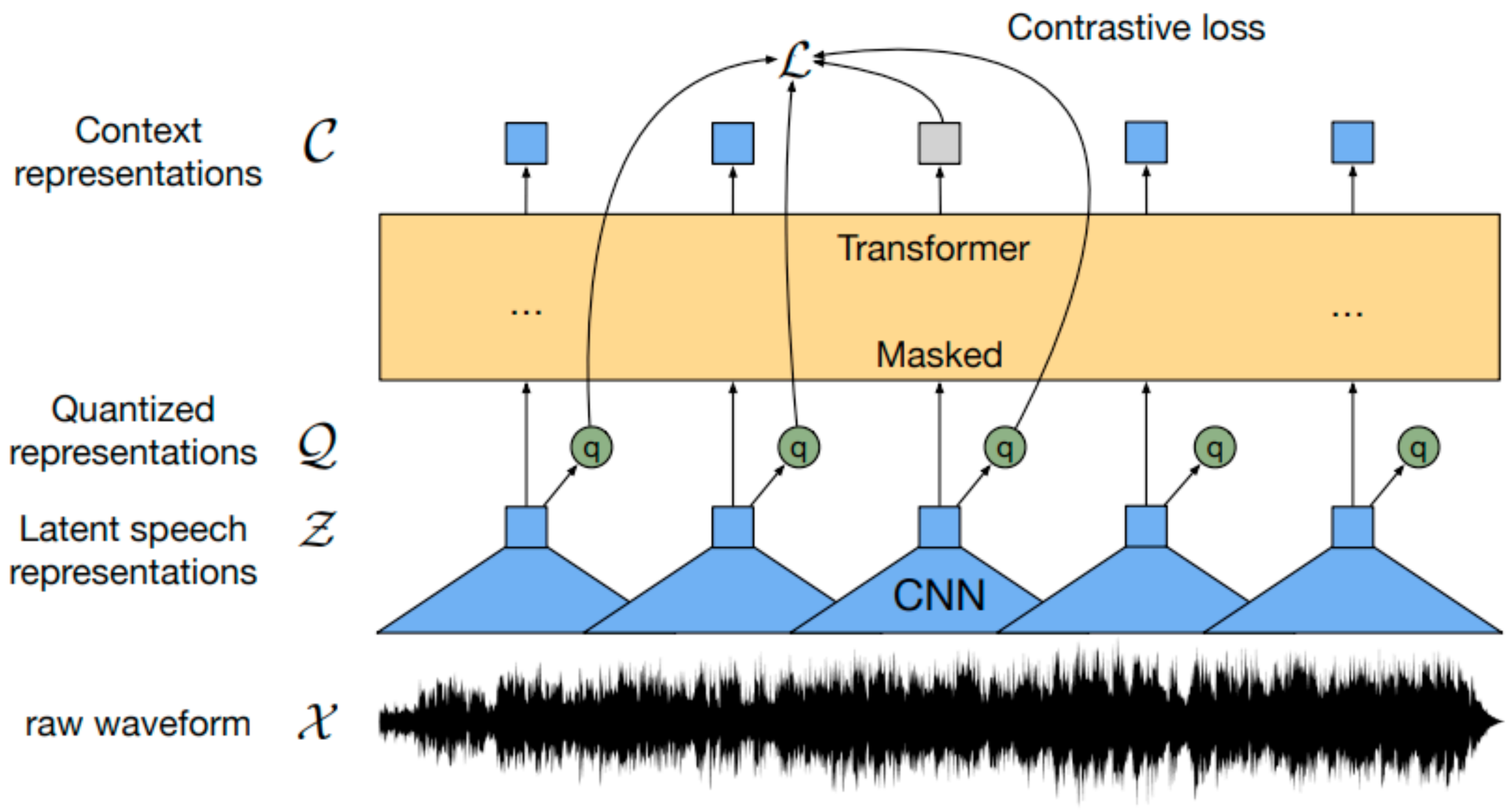
- A Language Model makes a **probabilistic prediction** about a **missing component** of a **sequence of symbols**
- "probabilistic": assigns a **probability** to **each possible prediction**
- "missing component": usually this is the **next symbol in the sequence**, given a certain prefix
  - BUT: some LMs predict a **missing word**, rather than the next one

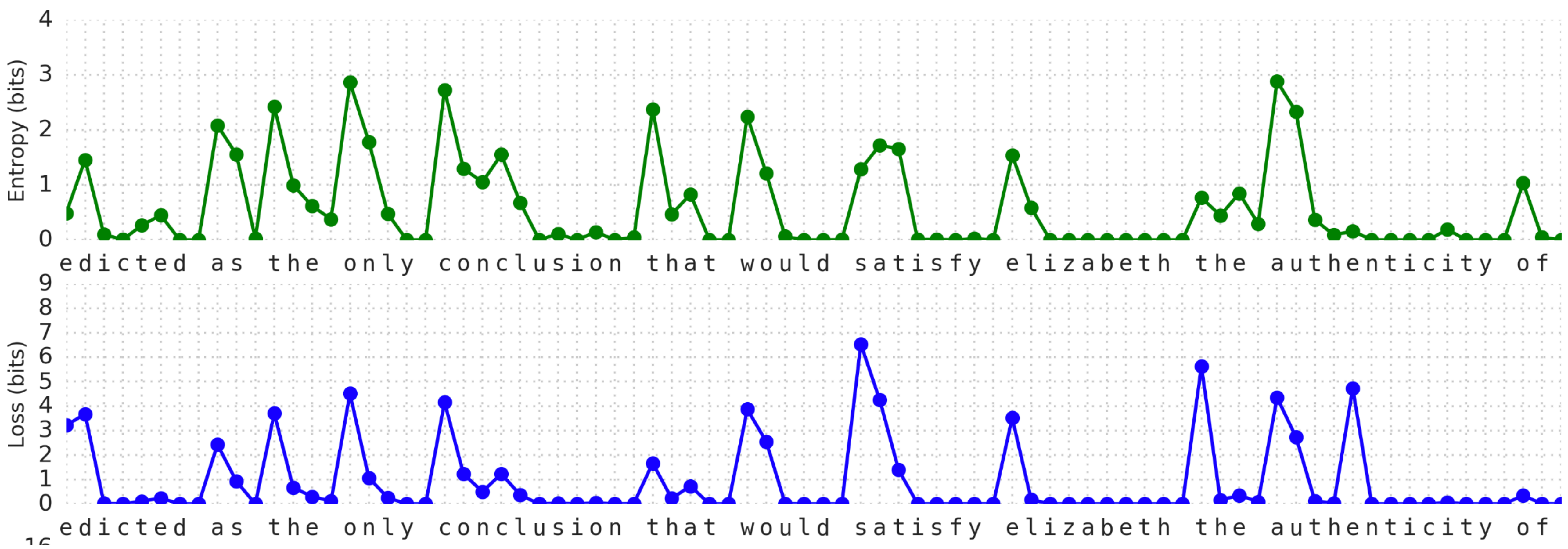
Language Modeling (Technical Definition)

- A Language Model makes a **probabilistic prediction** about a **missing component** of a **sequence of symbols**
- "probabilistic": assigns a **probability** to **each possible prediction**
- "missing component": usually this is the **next symbol in the sequence**, given a certain prefix
  - BUT: some LMs predict a **missing word**, rather than the next one
- "sequence of symbols": any **ordered sequence** of **discrete units**
  - Often words, but sometimes **characters, "sub-words", sound units, DNA**

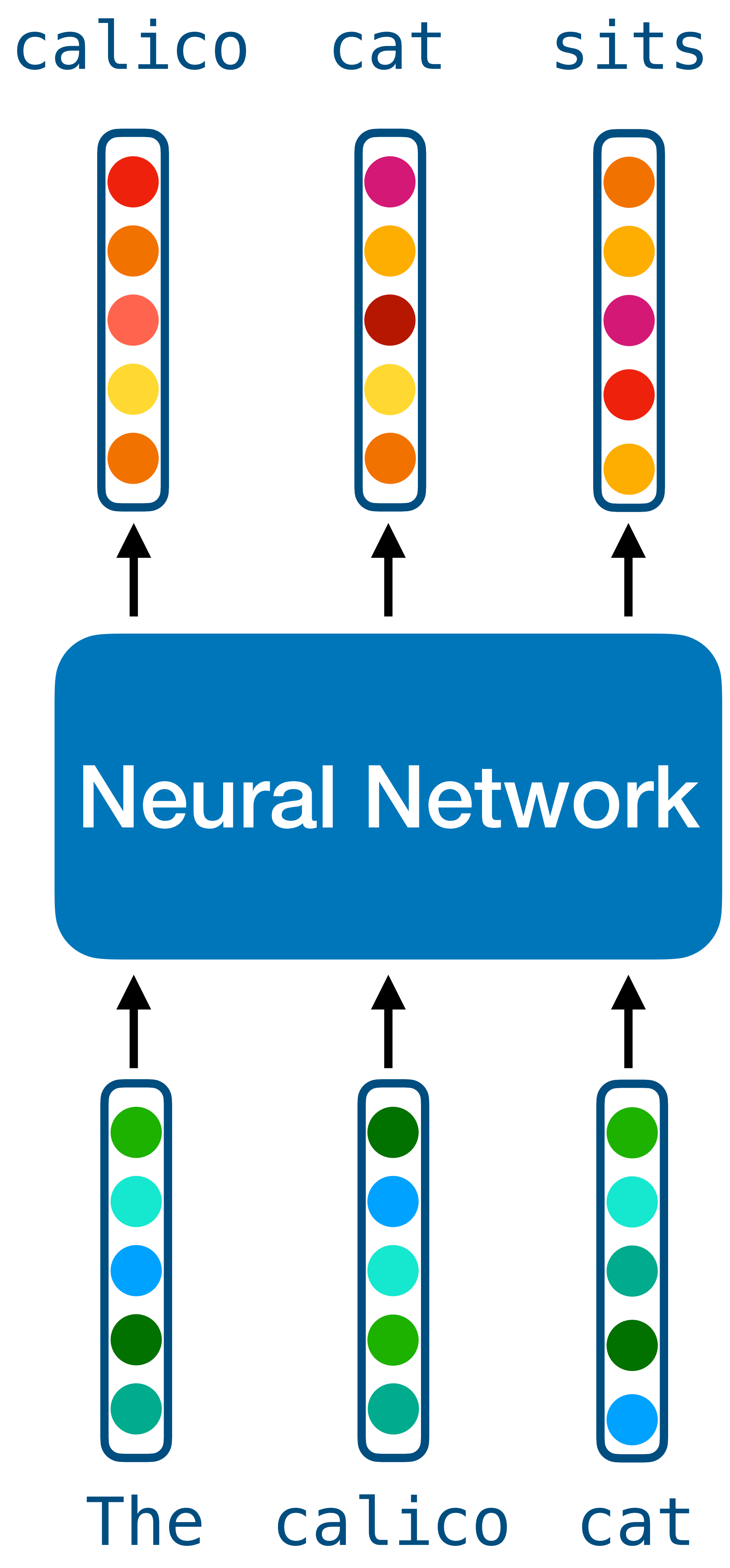
Language Models Without Words





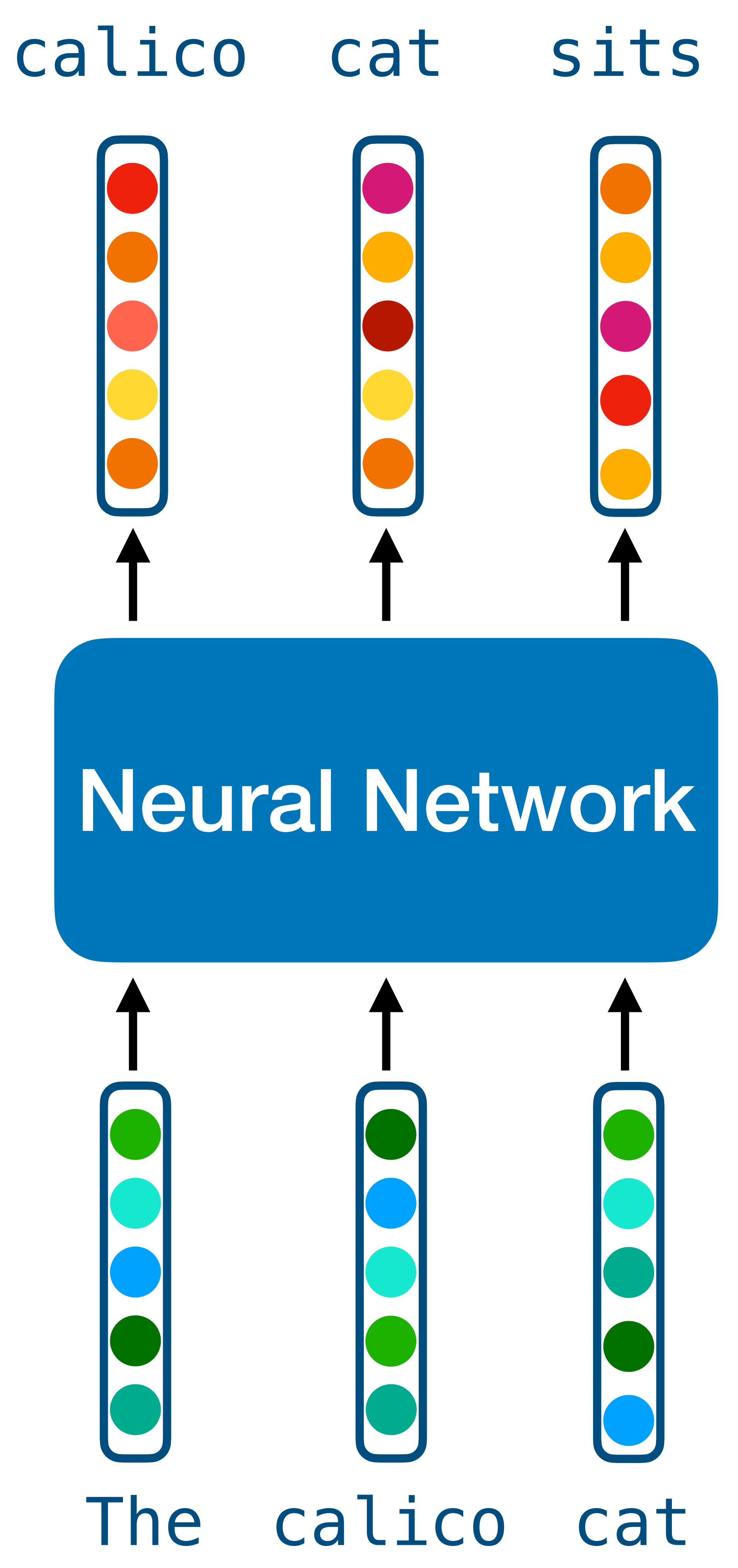


Neural Language Models



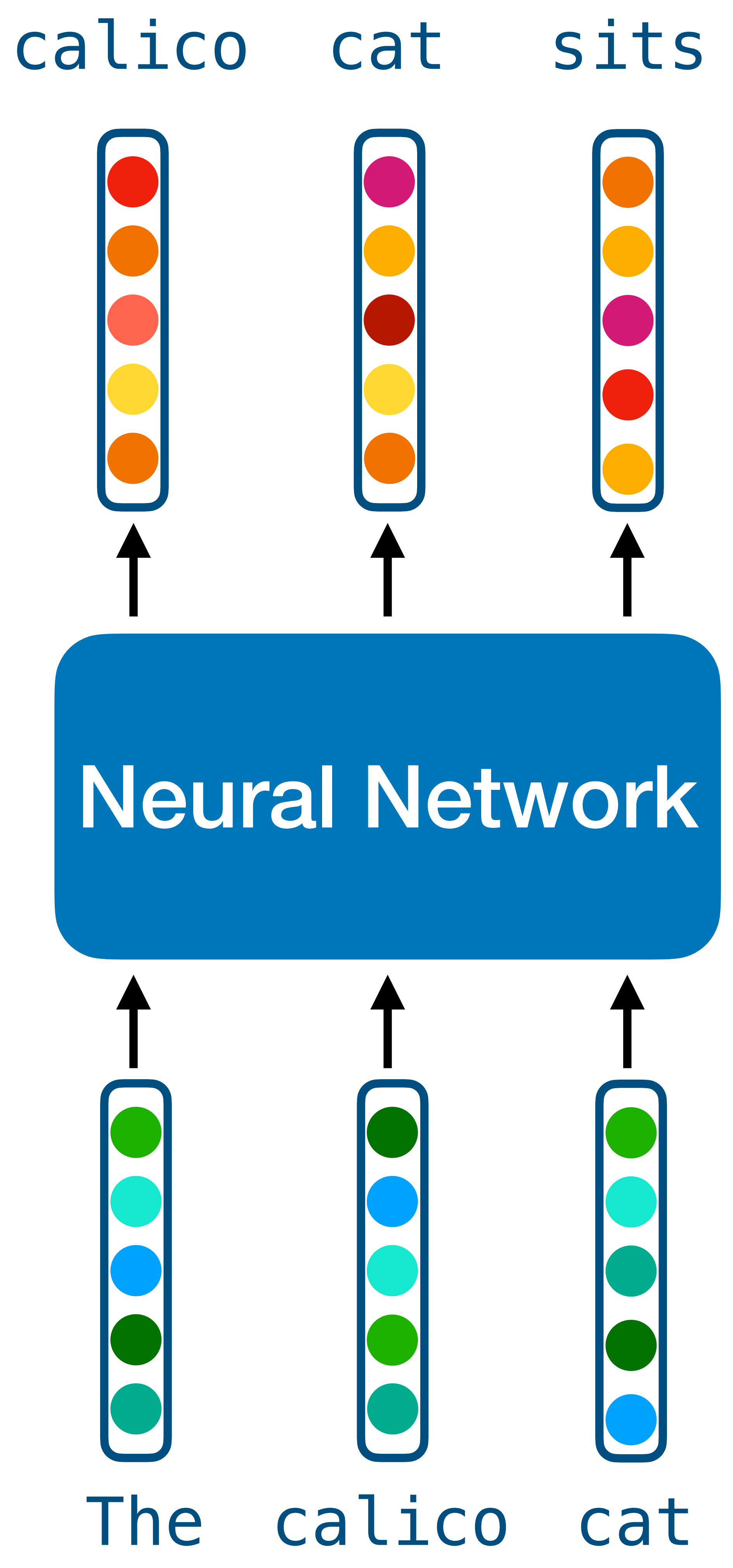
Neural Language Models

- As the name suggests: Neural Networks can be **trained to do Language Modeling**
  - All we need for this training is **language data** (text, audio, etc.)



Neural Language Models

- As the name suggests: Neural Networks can be **trained to do Language Modeling**
  - All we need for this training is **language data** (text, audio, etc.)
- **Vector representations** of language learned by the model are:
  - **Flexible/generalizable** to new instances, tasks
  - **Information-rich**, both in Linguistic structure and world knowledge



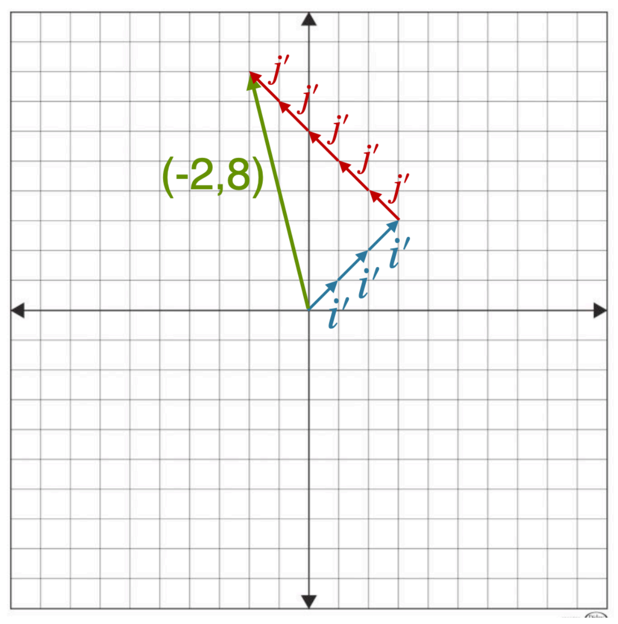
Course Structure

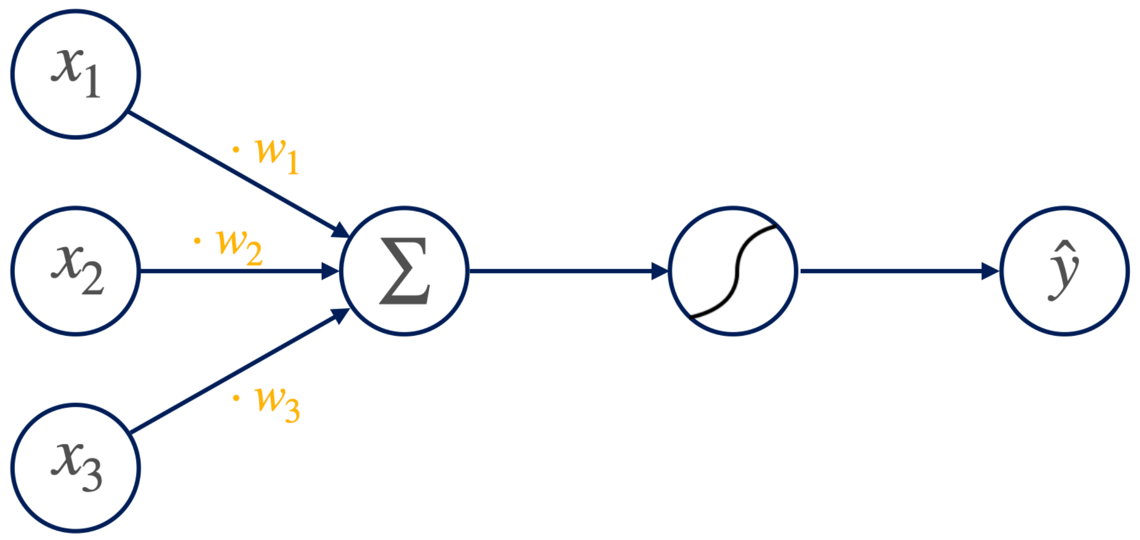
Overall Structure

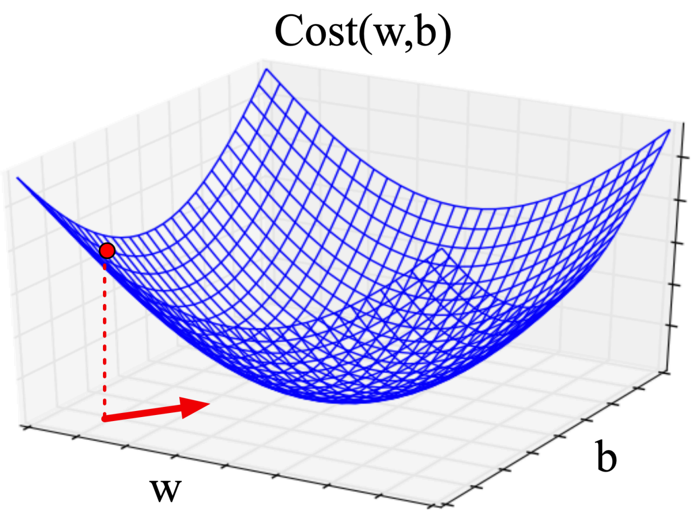
- Part 1: Mathematical and CompLing Foundations (~3 weeks)
- Part 2: Language Model Architectures (~3 weeks)
- Part 3: Special LM Topics (~4 weeks)
- Part 4: Speech Models (~1 week)
- Part 5: "Large Language Models" (~2 weeks)
- Part 6: Project Presentations and Wrap-up (~1 week)

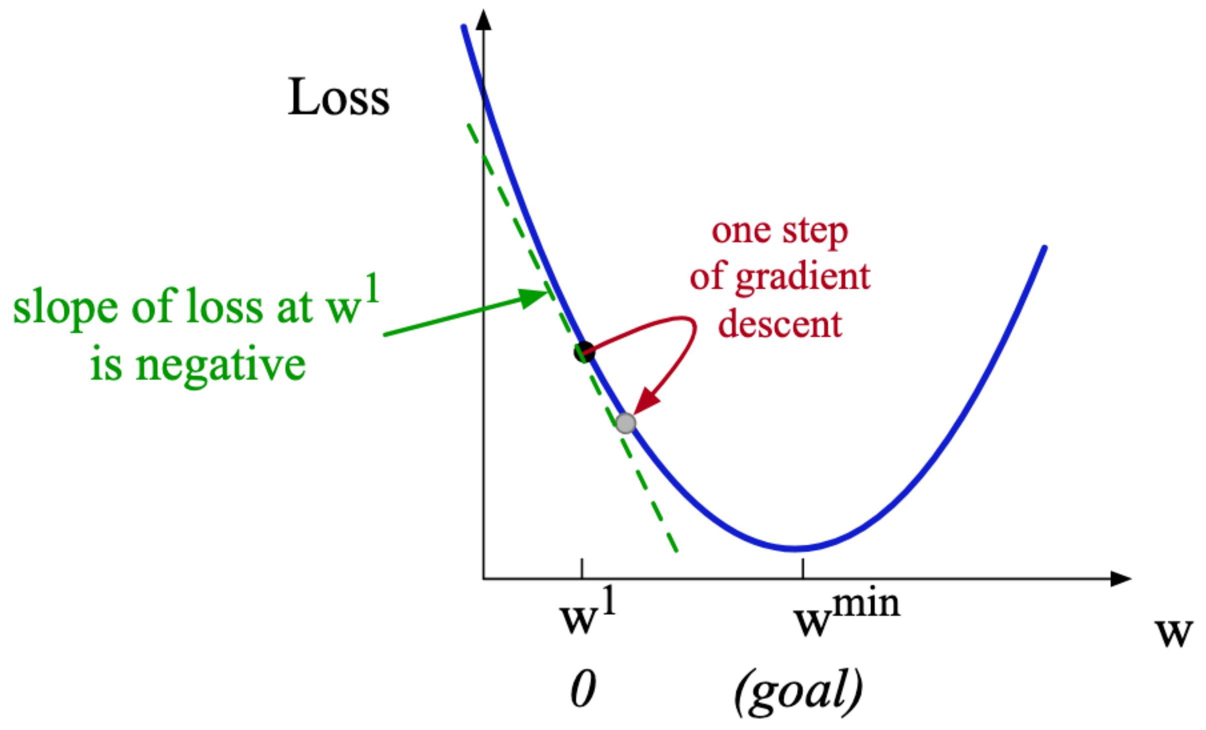
1: Math and CompLing Foundations

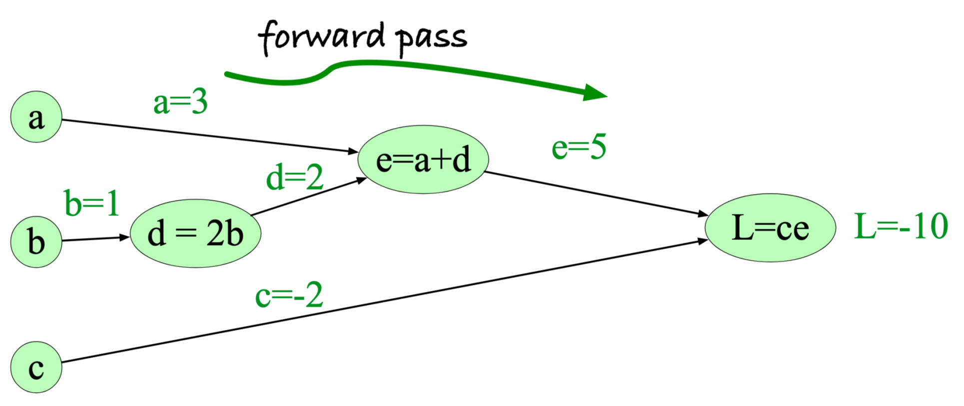
- Goal: build core knowledge in the underpinnings of Neural Networks
- Topics:
  - Vectors, Matrices, Linear Transformations
  - Supervised Learning, Gradient Descent
  - Computation Graphs
  - Artificial Neurons
  - Word Vectors
  - Probabilistic Language Modeling

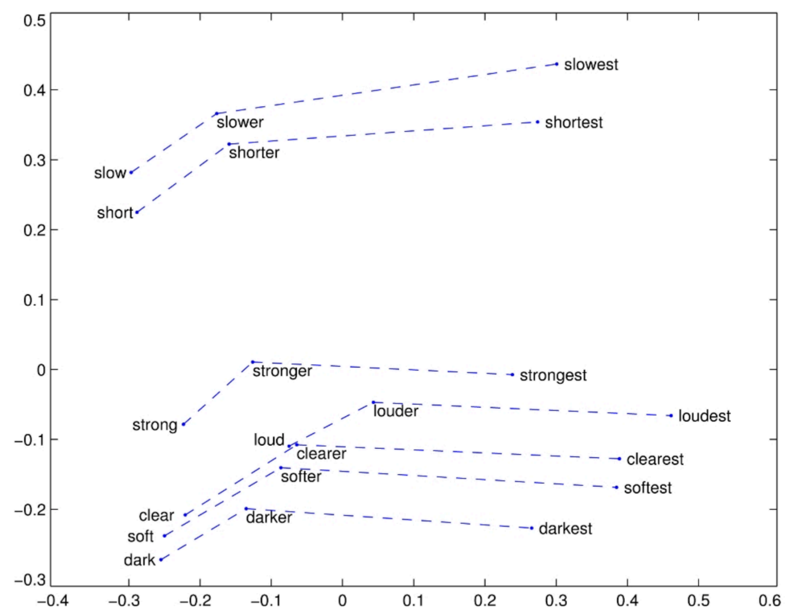






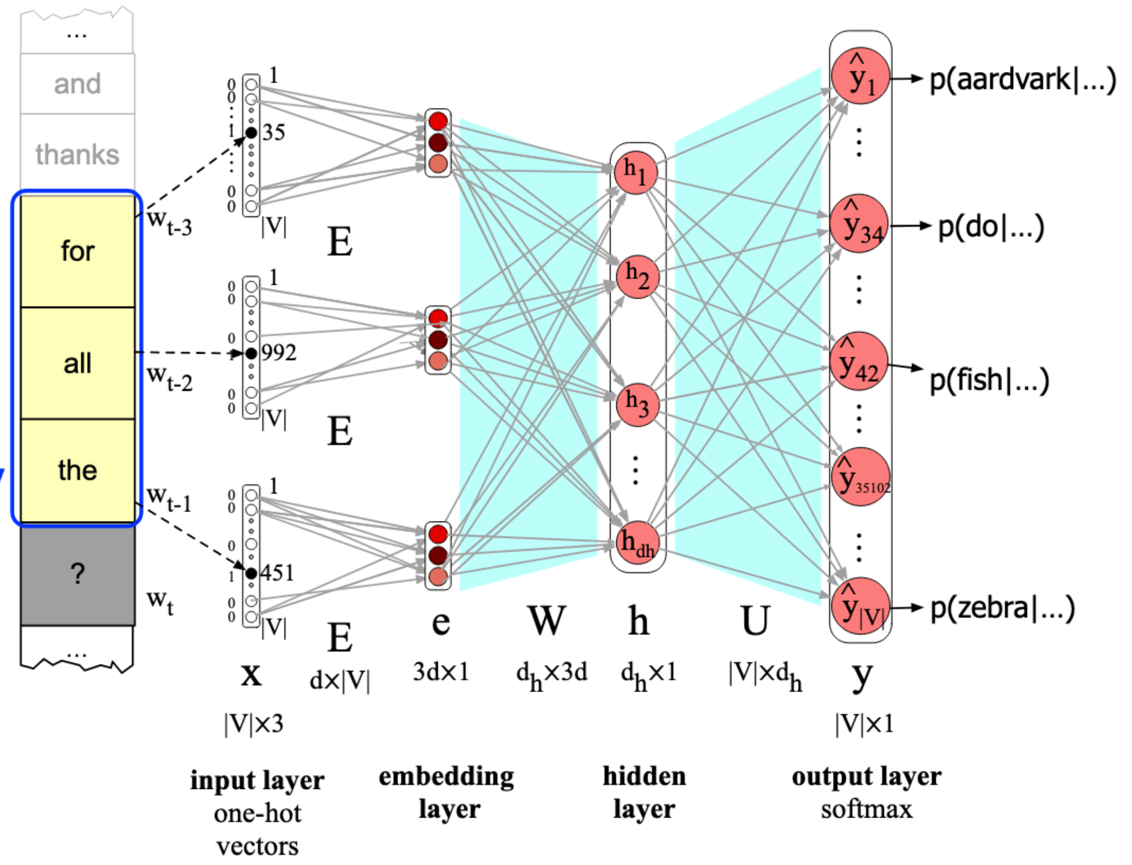


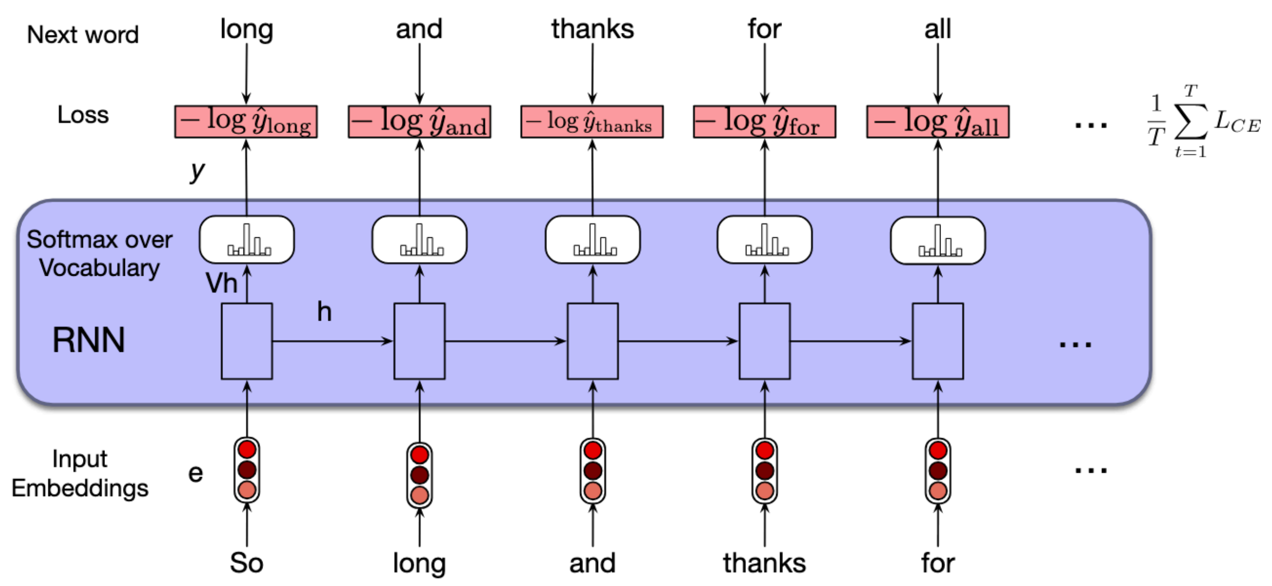


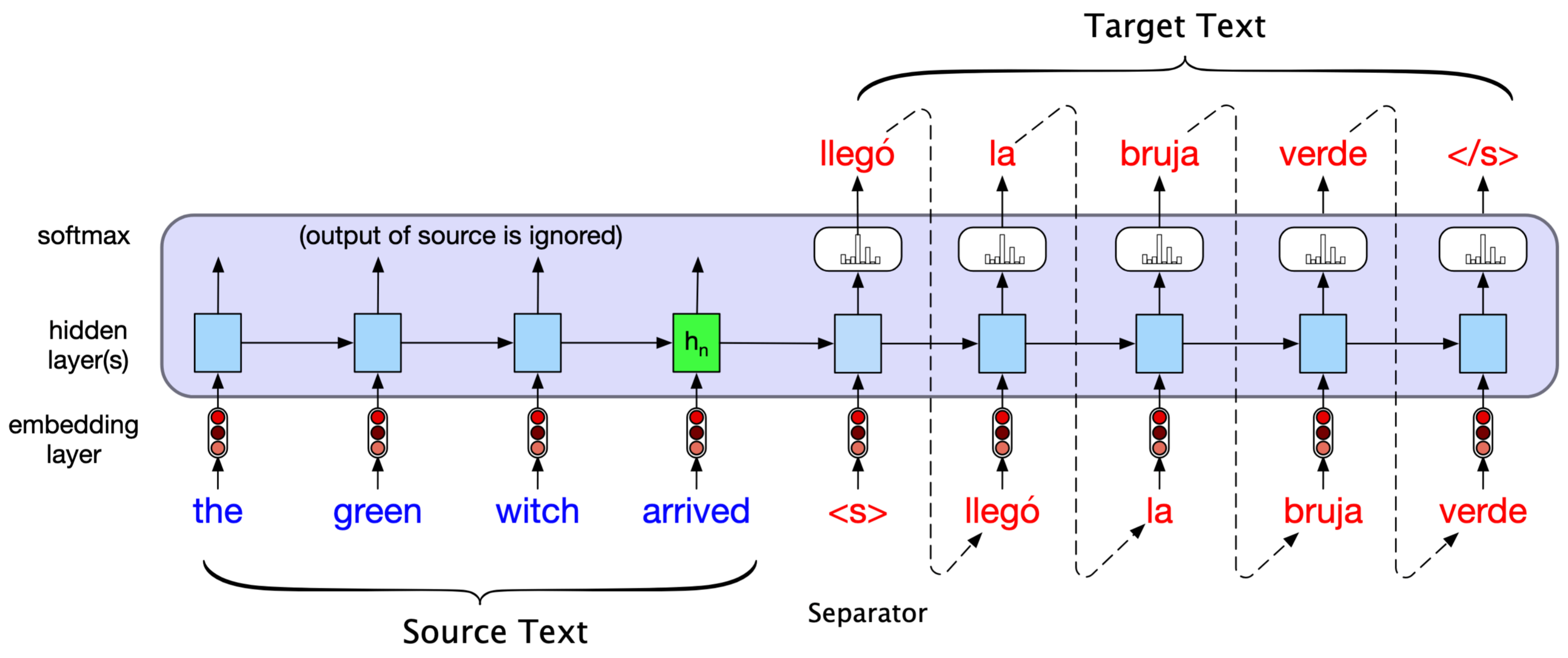


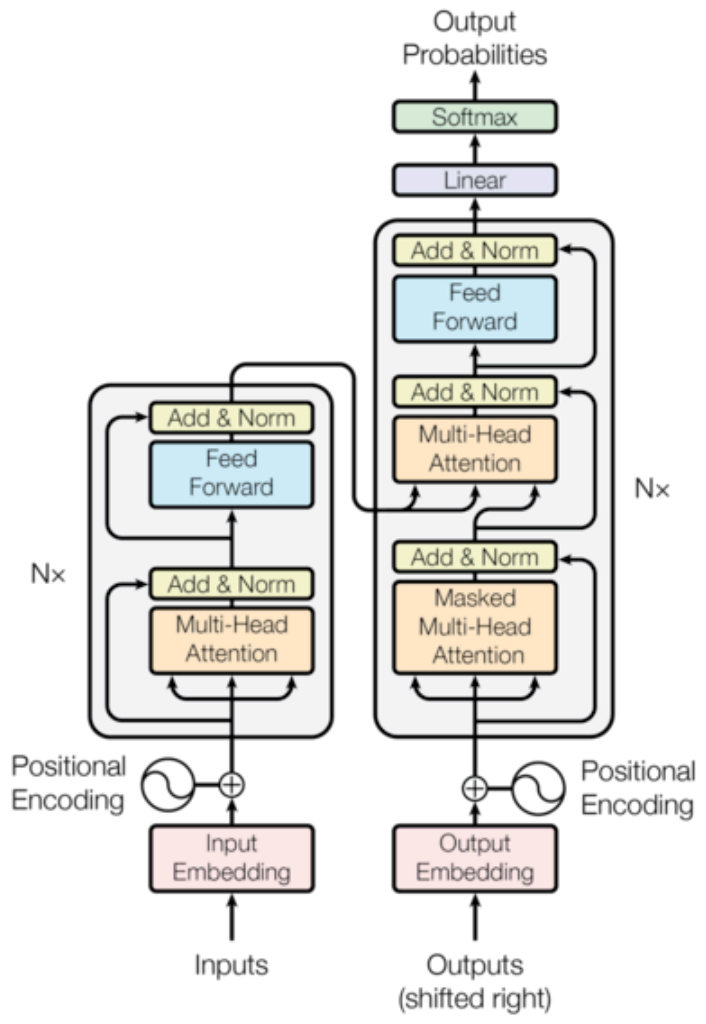
2: Language Model Architectures

- Goal: Survey different types of Neural Language Models
- Topics:
  - Feedforward NNs
  - Recurrent NNs
  - Recurrent variants / seq2seq
  - Transformers



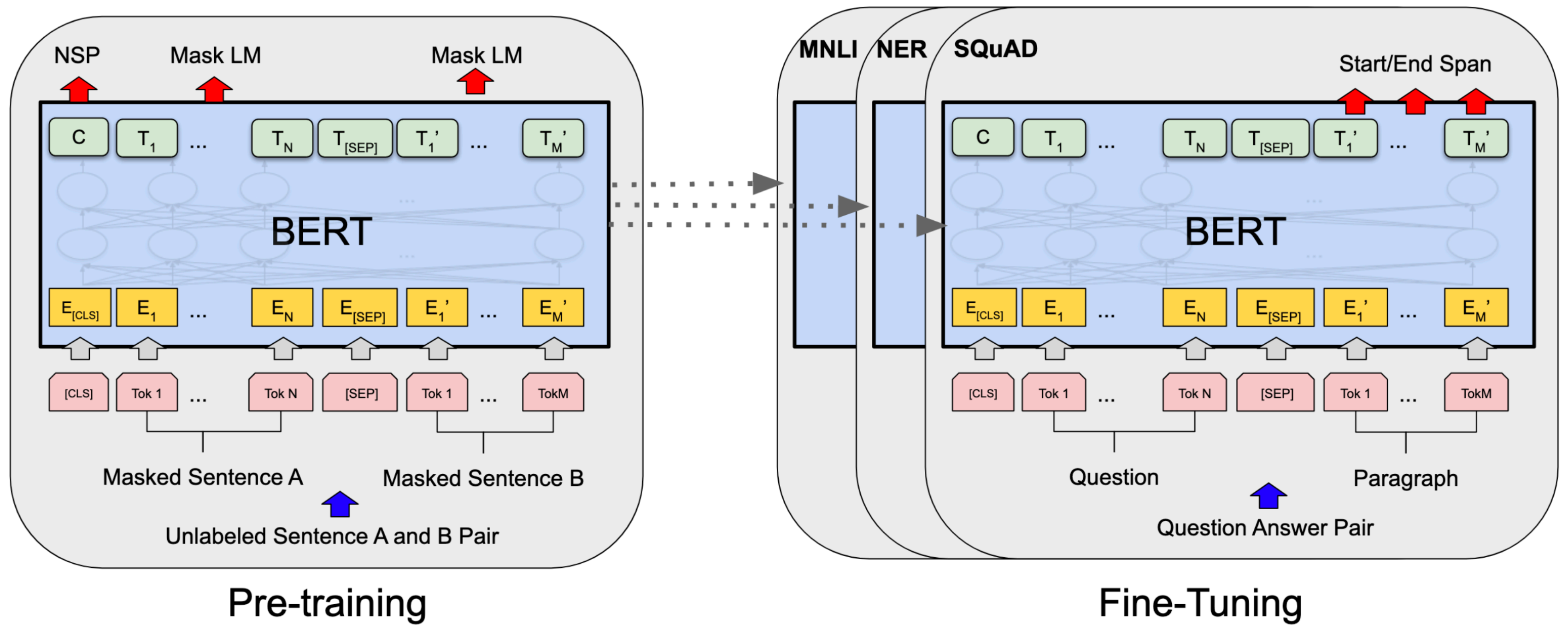


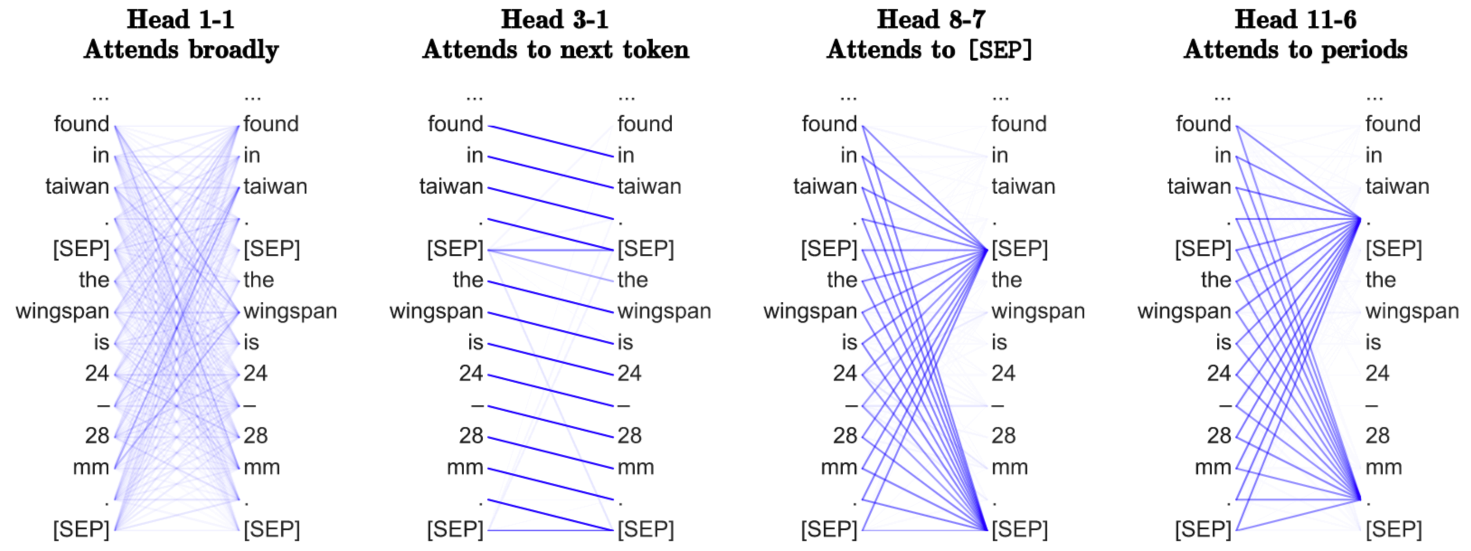


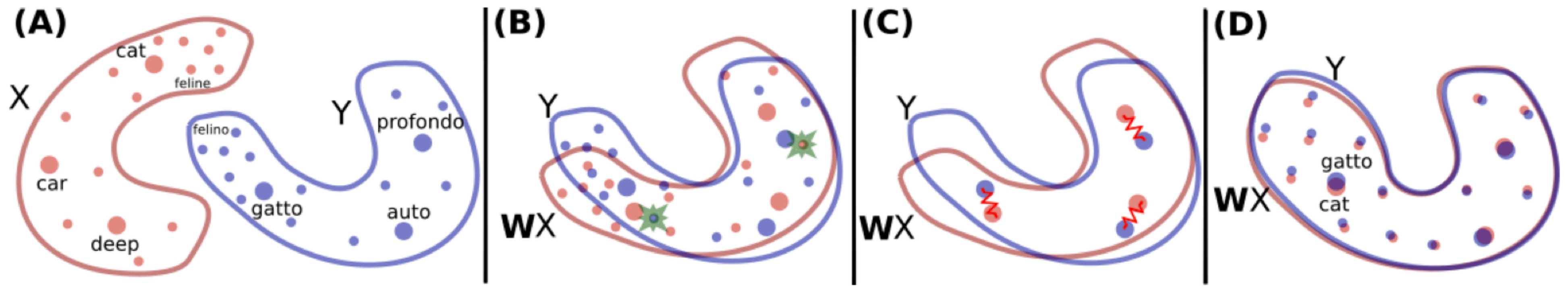


3: Special LM Topics

- Goal: Dig into specialized topics related to text LMs
- Topics:
  - Pre-training/Fine-tuning Paradigm
  - Decoding
  - Tokenization
  - Model Interpretability/Analysis
  - Multilingual Models

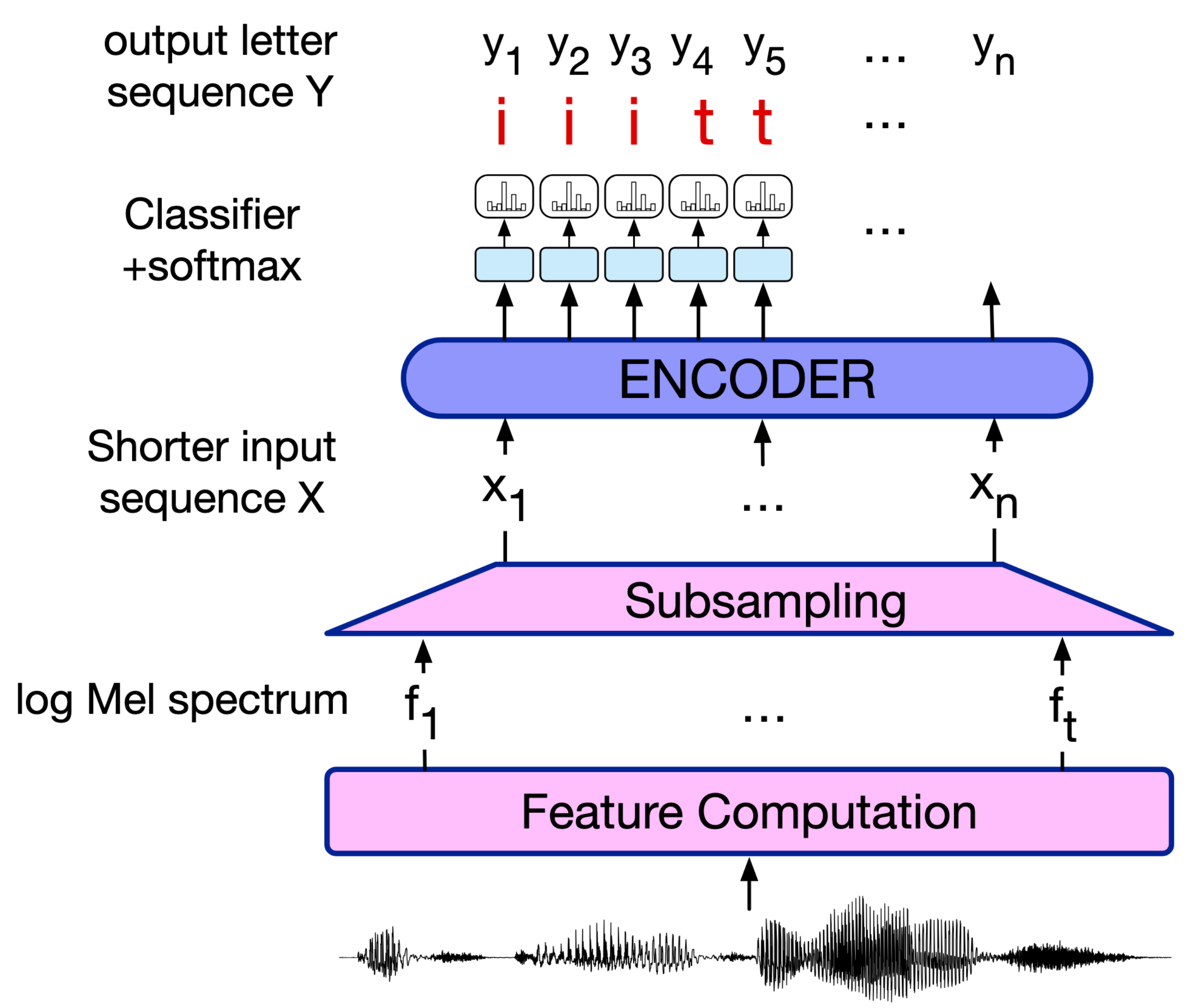


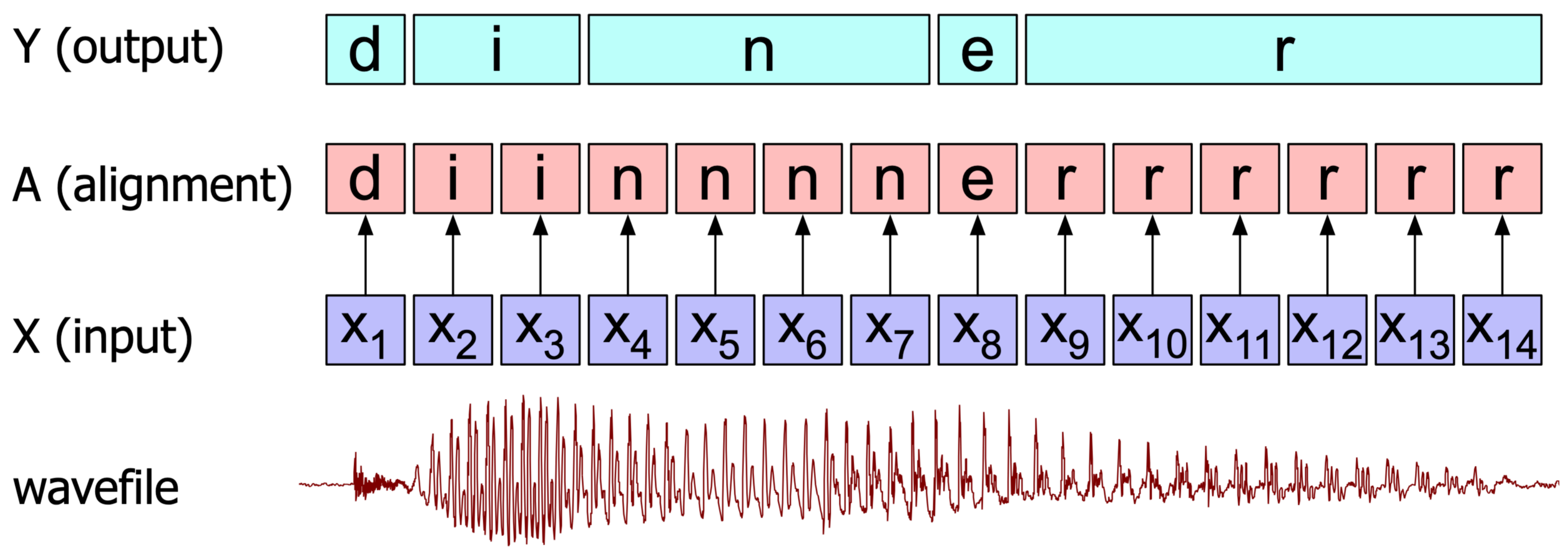




4: Speech Models

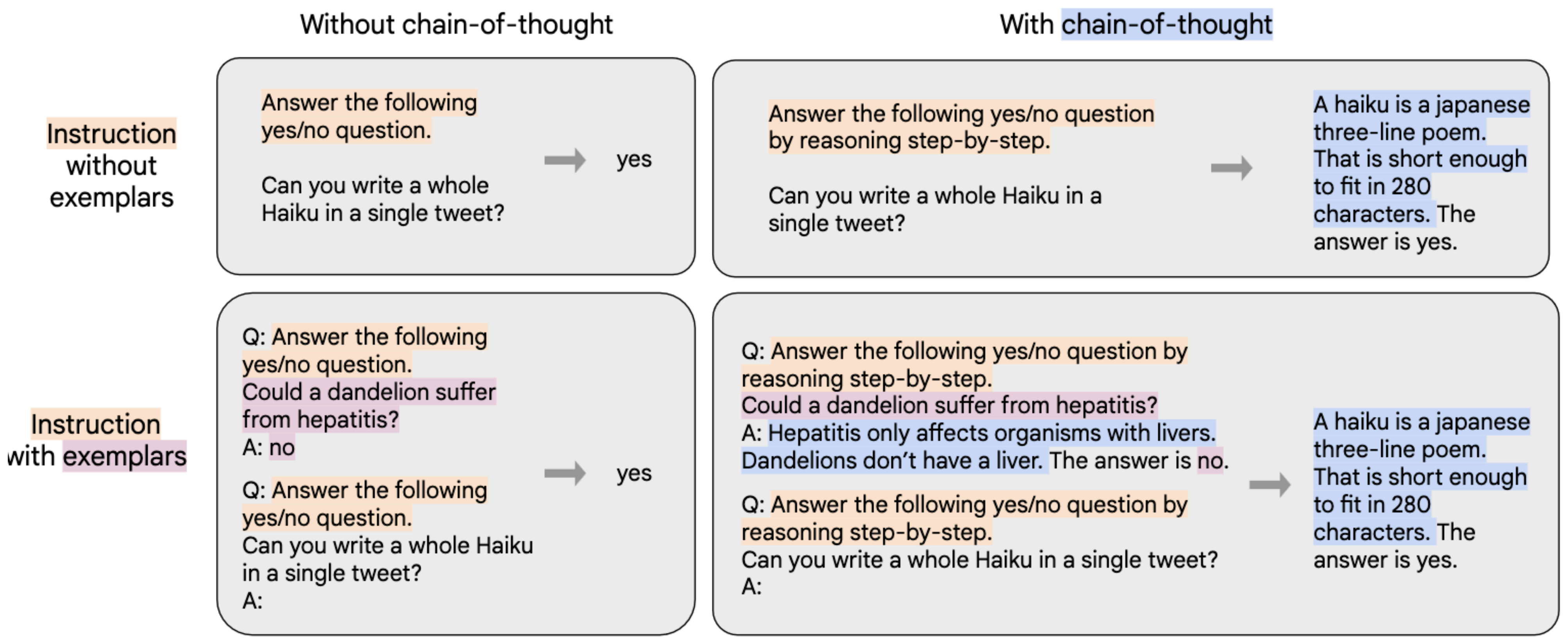
- Goal: Obtain a basic understanding of how speech-based LMs work
- Topics:
  - Acoustic Data
  - The Fourier Transform
  - CTC Loss
  - Speech LM Architectures

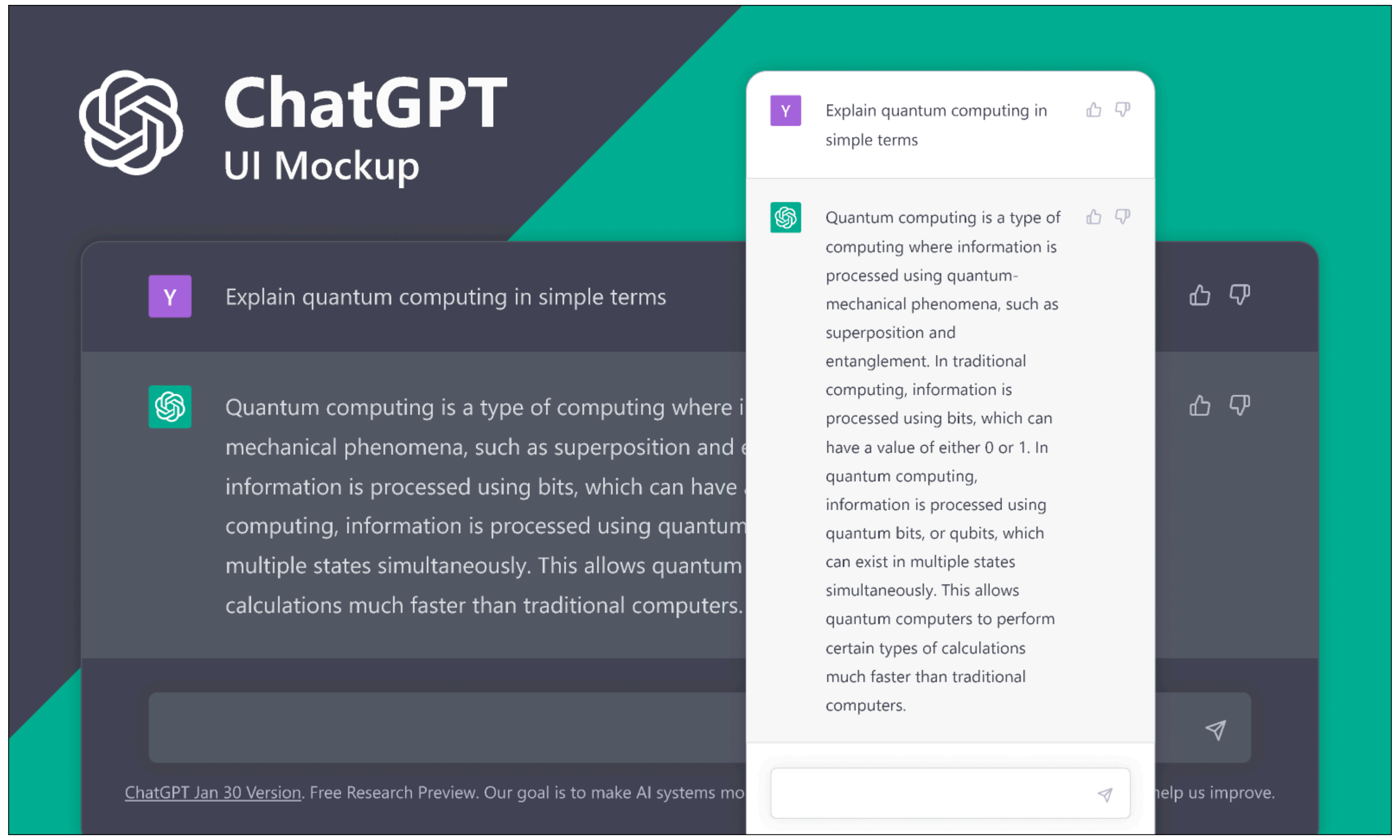




5: Large Language Models

- Goal: Understand innovations behind modern "Large Language Models" and how they relate to traditional Neural LMs
- Topics:
    - Prompting / In-context Learning
    - Alignment / Instruction Tuning
    - Reinforcement Learning
    - Potential Harms





What this course is **not**

What this course is **not**

- A course **all about "Large Language Models"**
  - We will **build up to LLMs** throughout the course. Almost **everything we learn will be useful** for understanding them
  - **CSC 511** focuses on LLMs exclusively

What this course is **not**

- A course **all about "Large Language Models"**
  - We will **build up to LLMs** throughout the course. Almost **everything we learn will be useful** for understanding them
  - **CSC 511** focuses on LLMs exclusively
- An overview of the **latest models and research**
  - We're building Neural Language Models **from the ground up**. That's a **70+ year history!**
  - You will have a chance to **engage with current research** in your term project!

What this course is **not**

- A course **all about "Large Language Models"**
  - We will **build up to LLMs** throughout the course. Almost **everything we learn will be useful** for understanding them
  - **CSC 511** focuses on LLMs exclusively
- An overview of the **latest models and research**
  - We're building Neural Language Models **from the ground up**. That's a **70+ year history!**
  - You will have a chance to **engage with current research** in your term project!
- **Software development-focused**, or a box of **"tips and tricks"**
  - This course focuses on **deep understanding** of neural models, which will help you understand more engineering-focused tips and tricks

Coursework / Grading

Grading

- Final grade composition
  - 40% homeworks
  - 30% term project
  - 20% in-class quizzes
  - 10% attendance & participation
- Late work penalty
  - 5% for 1st hour
  - 10% for 1st 24 hours
  - 20% for 1st 48 hours
  - **No grade** for work >48 hours late

Attendance

Attendance

- Attendance is **required** and I will keep track

Attendance

- Attendance is **required** and I will keep track
- You have **four "free" absences**
  - These can be used for **any reason** (including just not wanting to show up!)
  - You **do <u>not</u> have to contact me** to use these "free" absences
  - I will **apply these automatically**, showing as "excused" in Blackboard

Attendance

- Attendance is **required** and I will keep track
- You have **four "free" absences**
  - These can be used for **any reason** (including just not wanting to show up!)
  - You **do <u>not</u> have to contact me** to use these "free" absences
  - I will **apply these automatically**, showing as "excused" in Blackboard
- You may also request **formally excused** absences, which can be granted for important civic, religious, family reasons, etc.
  - These **don't count** towards your free absences if approved

Attendance

- Attendance is **required** and I will keep track
- You have **four "free" absences**
  - These can be used for **any reason** (including just not wanting to show up!)
  - You **do <u>not</u> have to contact me** to use these "free" absences
  - I will **apply these automatically**, showing as "excused" in Blackboard
- You may also request **formally excused** absences, which can be granted for important civic, religious, family reasons, etc.
  - These **don't count** towards your free absences if approved
- Make sure to **show up on quiz days!!**

Quizzes

Quizzes

- I will hold **in-class quizzes** on **most Mondays** (sometimes Wednesdays)
  - You will have the **first 15-20 minutes** of class to complete them
  - **Pen/pencil and paper** format! (I will bring some spare pencils)
  - All dates/topics found on the **course webpage**

Quizzes

- I will hold **in-class quizzes** on **most Mondays** (sometimes Wednesdays)
  - You will have the **first 15-20 minutes** of class to complete them
  - **Pen/pencil and paper** format! (I will bring some spare pencils)
  - All dates/topics found on the **course webpage**
- Designed to gauge **engagement with the material**, not to trick you!
  - Grades will be **shifted** so that the **median is 85%** (only shifted up)
  - **Example questions:** math, simple code snippets, short answer, multiple choice
  - I will **tell you beforehand** which skills/topics to expect on the quiz!

Homework

Homework

- 6-8 assignments with **written and programming** questions
  - Generally: questions that are **too long for quizzes**

Homework

- 6-8 assignments with **written and programming** questions
  - Generally: questions that are **too long for quizzes**
- Usually assigned on a Wednesday and **due in 1 week**
  - Deadline: **11pm Eastern Time** on the due date
  - Submitted on **Blackboard**

Homework

- 6-8 assignments with **written and programming** questions
  - Generally: questions that are **too long for quizzes**
- Usually assigned on a Wednesday and **due in 1 week**
  - Deadline: **11pm Eastern Time** on the due date
  - Submitted on **Blackboard**
- Use of generative AI is **not allowed** on these homeworks

Term Project

Term Project

- Over the term you will complete a **group research project**, which must:
  - Answer a **scientific question** / hypothesis about **language or linguistic theory**
  - Employ a **neural model of language** which you have **trained or fine-tuned**
  - Address a **creative topic** based on group interests

Term Project

- Over the term you will complete a **group research project**, which must:
  - Answer a **scientific question** / hypothesis about **language or linguistic theory**
  - Employ a **neural model of language** which you have **trained or fine-tuned**
  - Address a **creative topic** based on group interests
- The project will culminate in a **scientific-style paper** and presentation

Term Project

- Over the term you will complete a **group research project**, which must:
  - Answer a **scientific question** / hypothesis about **language or linguistic theory**
  - Employ a **neural model of language** which you have **trained or fine-tuned**
  - Address a **creative topic** based on group interests
- The project will culminate in a **scientific-style paper** and presentation
- You will be given access to UR's **BlueHive Computing Cluster** to develop your model and run experiments

Term Project cont.

Term Project cont.

- You will also complete **project milestones** throughout the semester to ensure your topic is **feasible** and **on-track**
  - These will count toward your term project grade

Term Project cont.

- You will also complete **project milestones** throughout the semester to ensure your topic is **feasible** and **on-track**
  - These will count toward your term project grade
- Periodic **self-assessments** will ensure work is **distributed fairly**
  - I understand that everyone will have **different skill-sets**
  - Each group should consciously plan **how best to distribute tasks**
  - Project milestones will be aimed at helping with this

Resources

Webpage

- https://cmdowney88.github.io/teaching/ling282/fall25
  - Up-to-date **schedule**
  - **Lecture slides** posted
  - **Homeworks** posted
  - Links to **required readings**
  - Course **info** and **policies**
- Always check here for **important dates!**

LING 282/482: Deep Learning in Computational Linguistics (Fall '25) — Information Policies Schedule

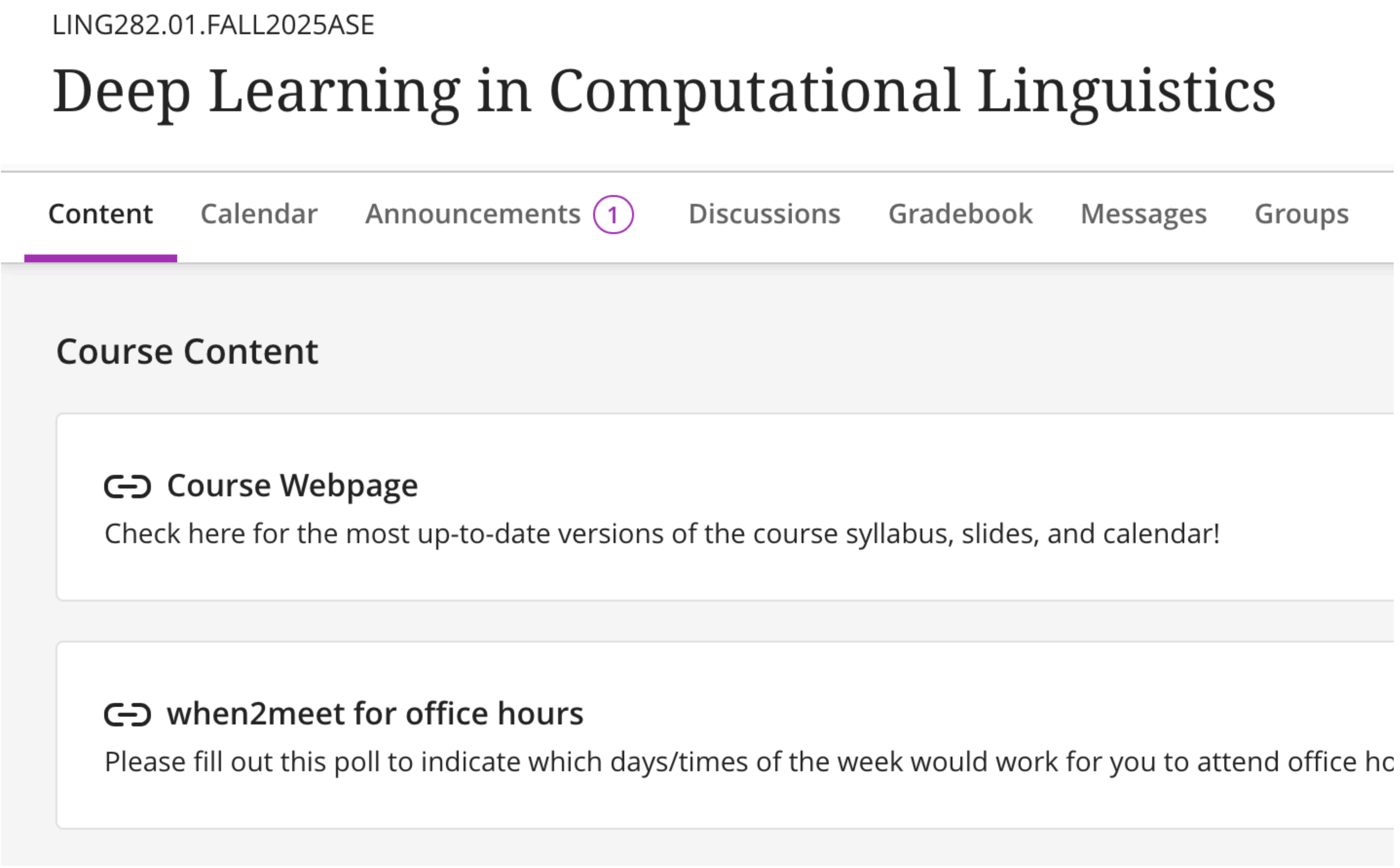
Schedule

(subject to change)

Date	Topics + Slides	Required Readings	Events
Aug 25	Introduction, Deep Learning History		
Aug 27	Vectors and Linear Transformations	Essence of Linear Algebra Ch.1-8	
Sep 1		Labor Day: no class	
Sep 3	The Perceptron	JM 4.0, 5.0-5.2.2	**in-class quiz:** vectors, matrices, linear transformations
Sep 8	Supervised Learning, Gradient Descent	JM 5.4-5.6	**in-class quiz:** function derivatives (Calc. pre-reqs)

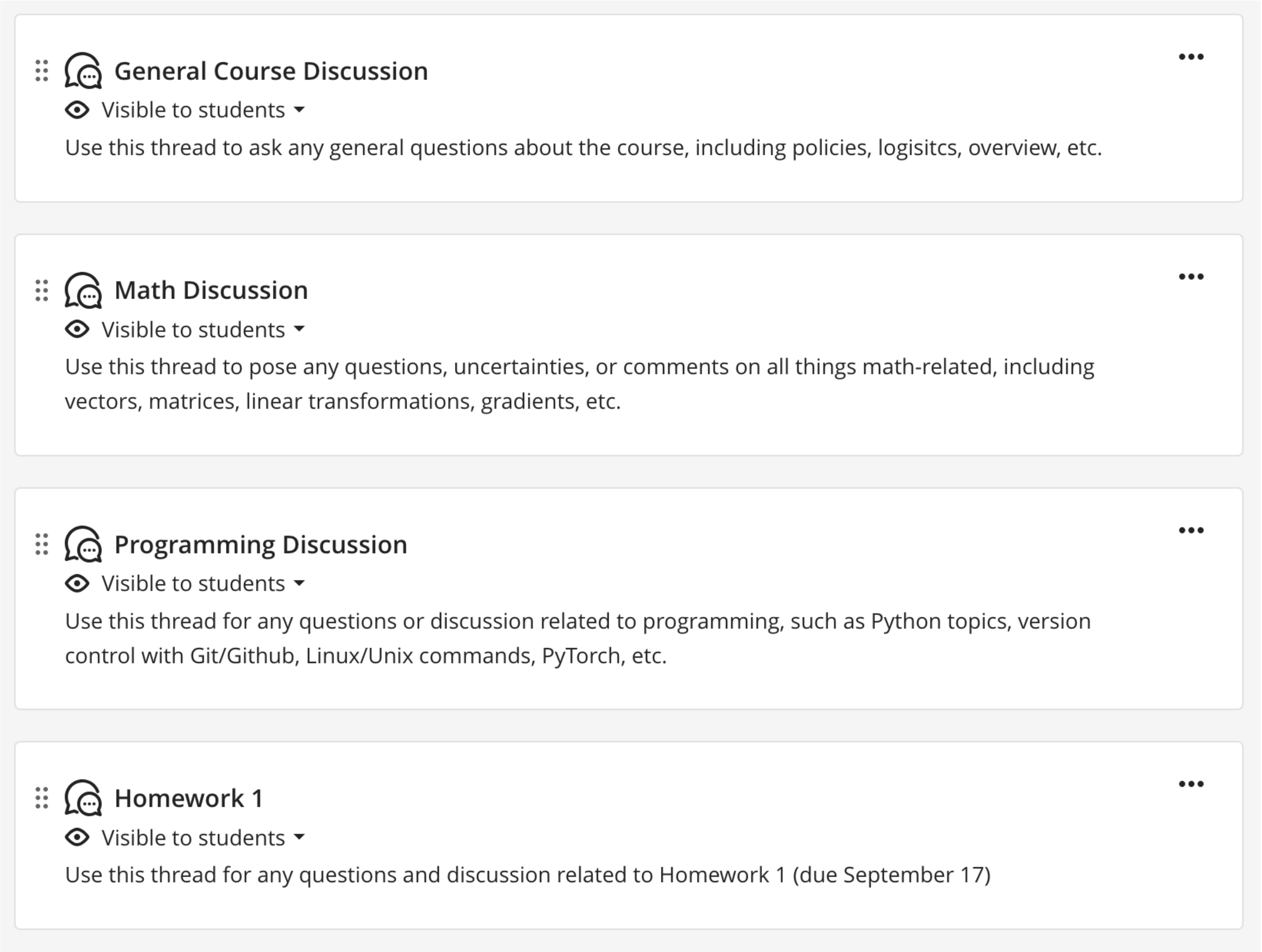
Blackboard

- We will use Blackboard for:
  - **Submitting assignments**
  - **Announcements**
  - **Discussion boards**
  - **Grades**
  - **Messaging** (optional, you can also email me)
- There is also a **link to the webpage** at the top of the Blackboard homepage



Blackboard Discussions

- Please make use of the **Blackboard Discussions section!**
- Feel free to post **any course-related questions** or comments
  - The thread topics are loose guides to organize discussion
  - If you have a question, **someone else might the same one!**
- I will try to respond to the discussion boards **within business hours**
  - **Email me** for **urgent** questions, but don't expect replies within hours of a due date :)



Contact and Office Hours

- Email: c.m.downey@rochester.edu
- Office hours
  - Time TBD: please fill out the **when2meet on Blackboard!**
    - I will try to pick a time when most people are available
  - Lattimore 507
  - Drop-in (no need to make an appointment)

Questions?

A Short History of NNs

The first artificial neural network: 1943

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE
IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH AND WALTER PITTS

FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

Turing Award: 2018

acm

MORE ACM AWARDS

A.M. TURING AWARD

A.M. TURING AWARD WINNERS BY...

ALPHABETICAL LISTING | YEAR OF THE AWARD | RESEARCH SUBJECT

Yoshua Bengio

Geoffrey E Hinton

Yann LeCun

GEOFFREY HINTON AND YANN LECUN TO DELIVER TURING LECTURE AT FCRC 2019

June 23, 5:15 - 6:30 P.M., Symphony Hall

We are pleased to announce that Geoffrey Hinton and Yann LeCun will deliver the Turing Lecture at FCRC 2019. Hinton's talk, "The Deep Learning Revolution," and LeCun's talk, "The Deep Learning Revolution: The Sequel," will be presented June 23rd from 5:15-6:30pm in Symphony Hall, Phoenix, Arizona.

No registration or tickets necessary to attend.

View the Livestream

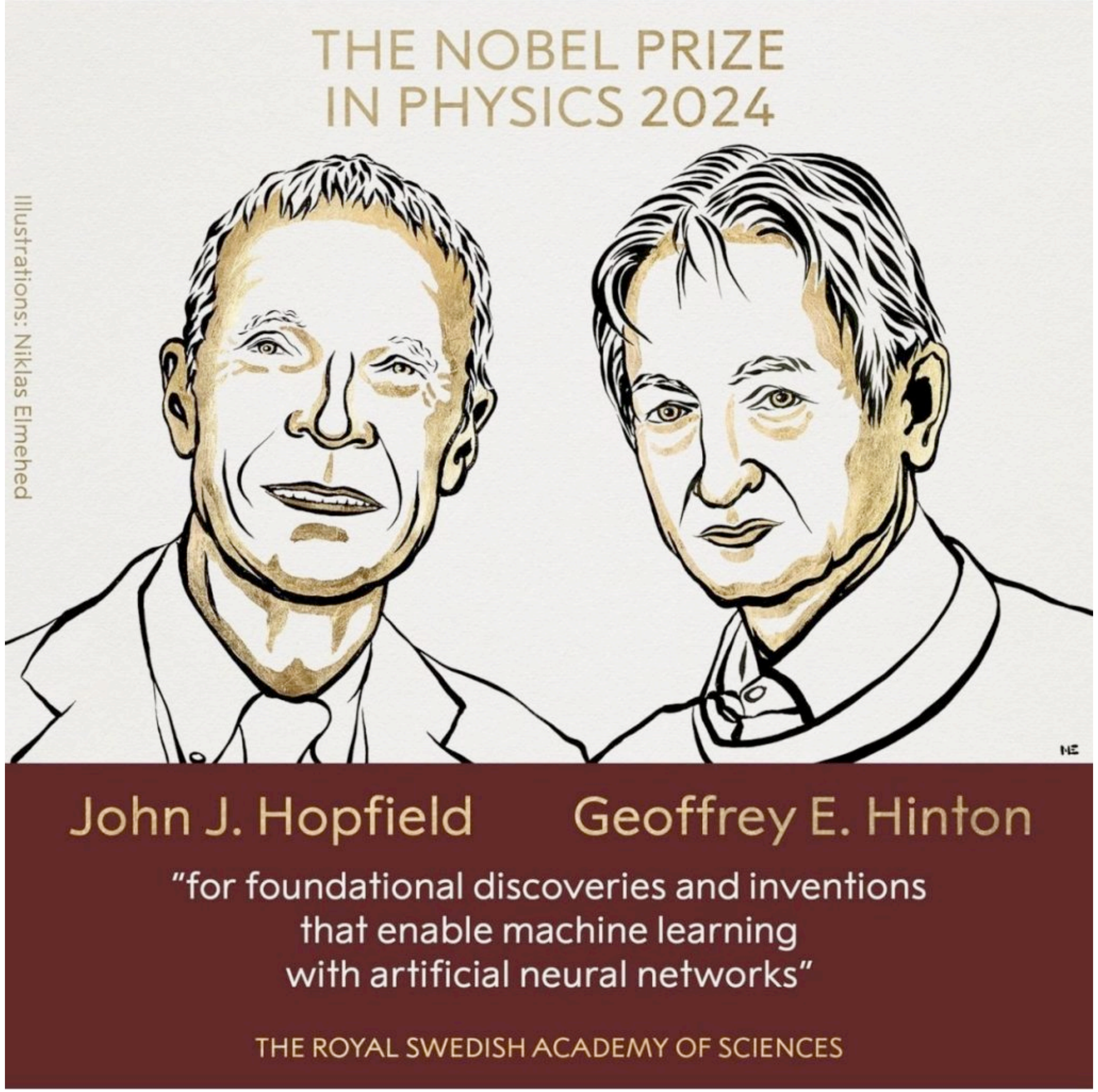
FATHERS OF THE DEEP LEARNING REVOLUTION RECEIVE ACM A.M. TURING AWARD

Bengio, Hinton, and LeCun Ushered in Major Breakthroughs in Artificial Intelligence

ACM named Yoshua Bengio, Geoffrey Hinton, and Yann LeCun recipients of the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing. Bengio is Professor at the University of Montreal and Scientific Director at Mila, Quebec's Artificial Intelligence Institute; Hinton is VP and Engineering Fellow of Google, Chief Scientific Adviser of The Vector Institute, and University Professor Emeritus at the University of Toronto; and LeCun is Professor at New York University and VP and Chief AI Scientist at Facebook.

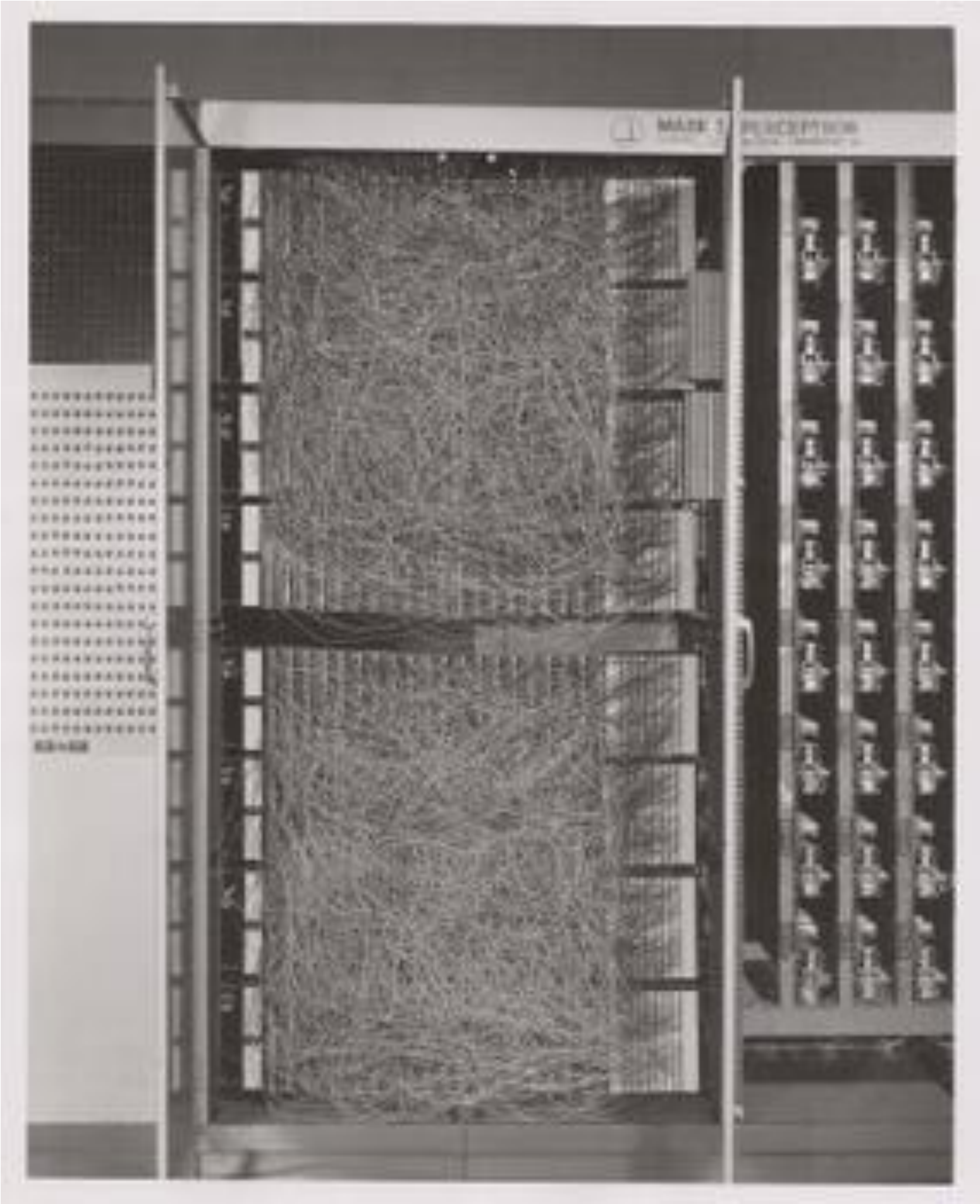
Working independently and together, Hinton, LeCun and Bengio developed conceptual foundations for the field, identified surprising phenomena through experiments, and contributed engineering advances that demonstrated the practical advantages of deep neural networks. In recent years, deep learning methods have been

Nobel Award: 2024



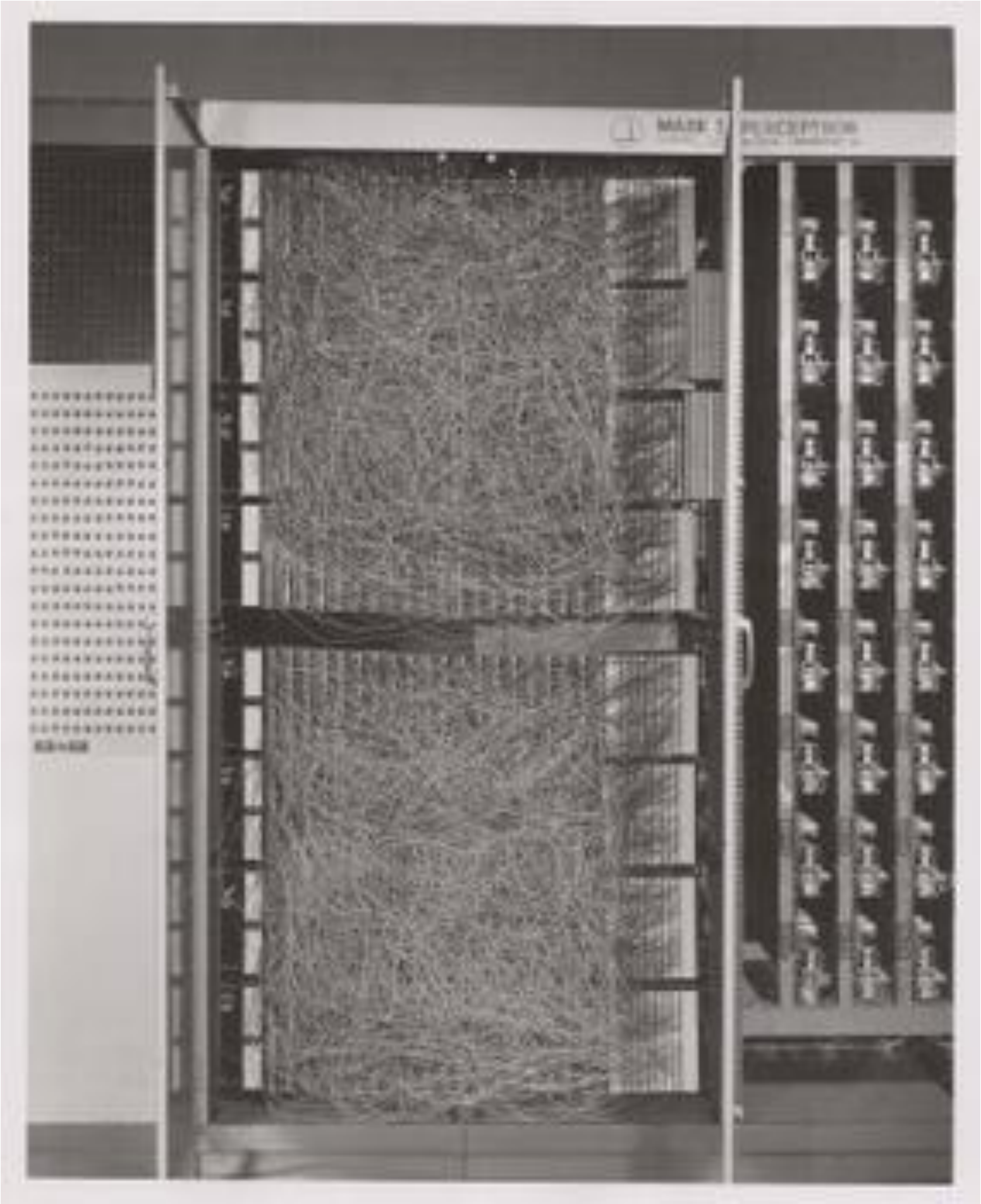
What took so long?

Perceptron (1958)



source

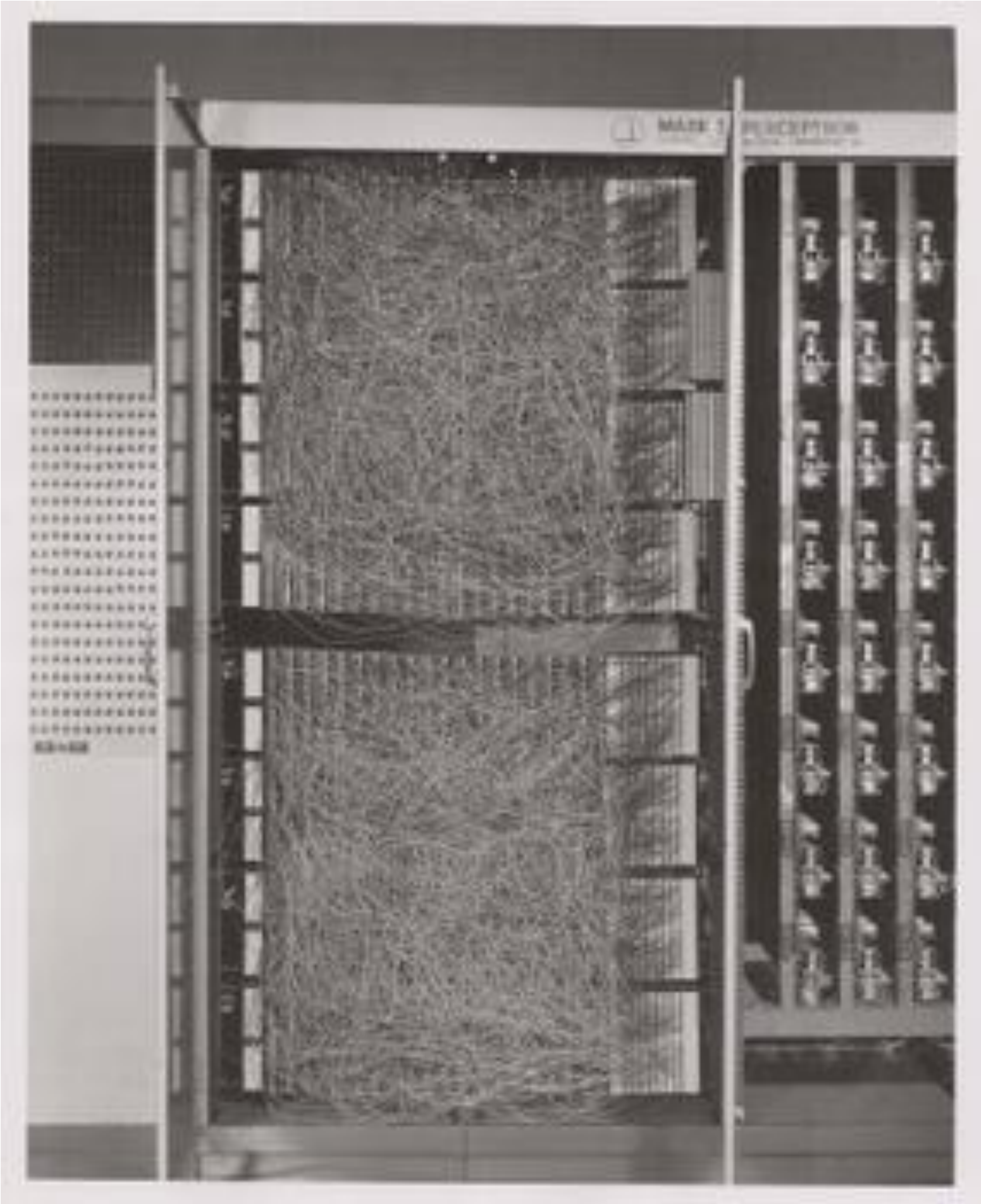
Perceptron (1958)



source

$$f(\mathbf{x}) = \begin{cases} 1 & \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Perceptron (1958)



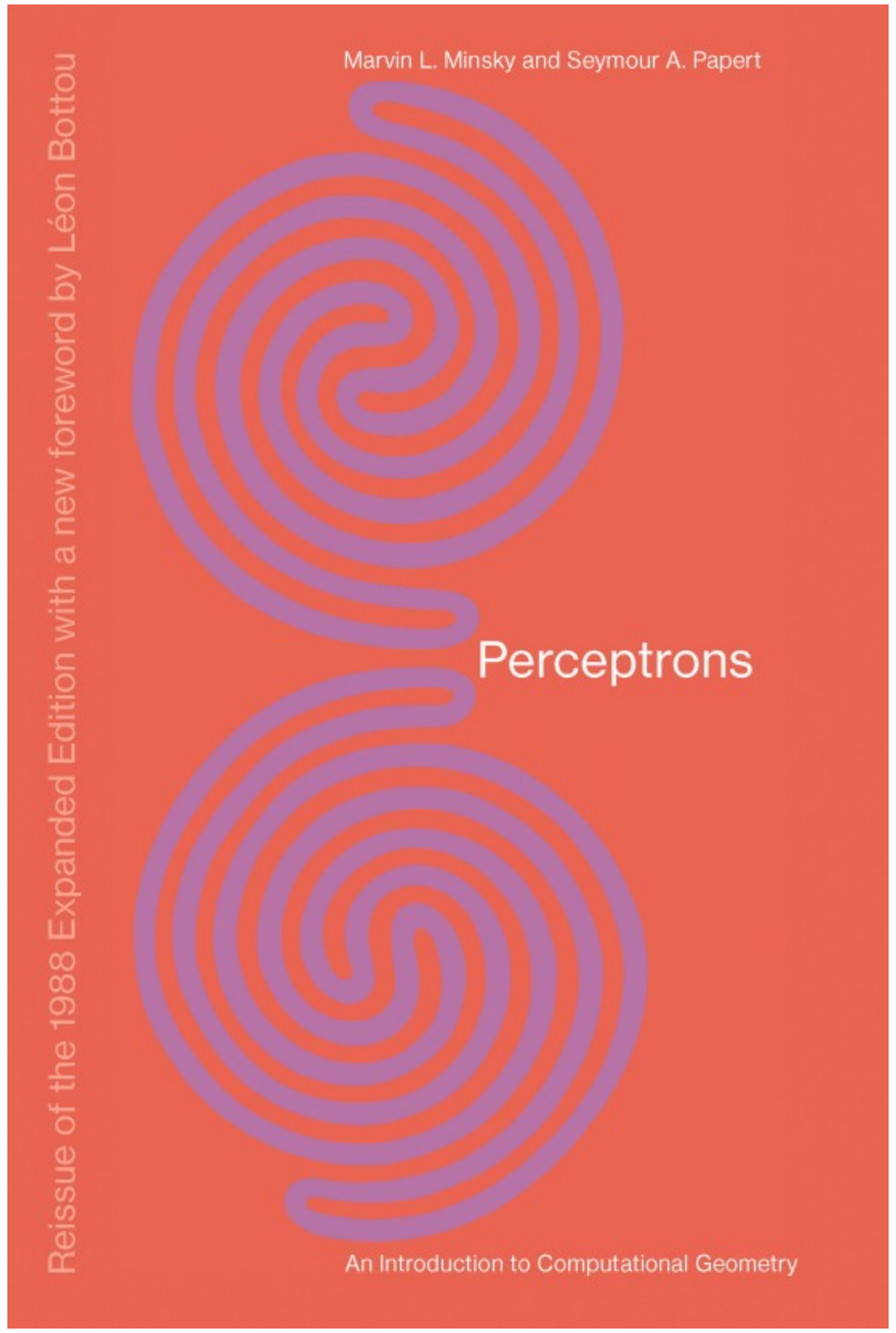
source

$$f(\mathbf{x}) = \begin{cases} 1 & \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

"the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."
—New York Times

Perceptrons (1969)

- Shows that Perceptrons are **limited** in what **kind of functions** they can compute
- Famous example: **Exclusive Disjunction** (XOR)
  - Not computable by a (single) Perceptron
  - We'll return to this



"AI Winter"

"AI Winter"

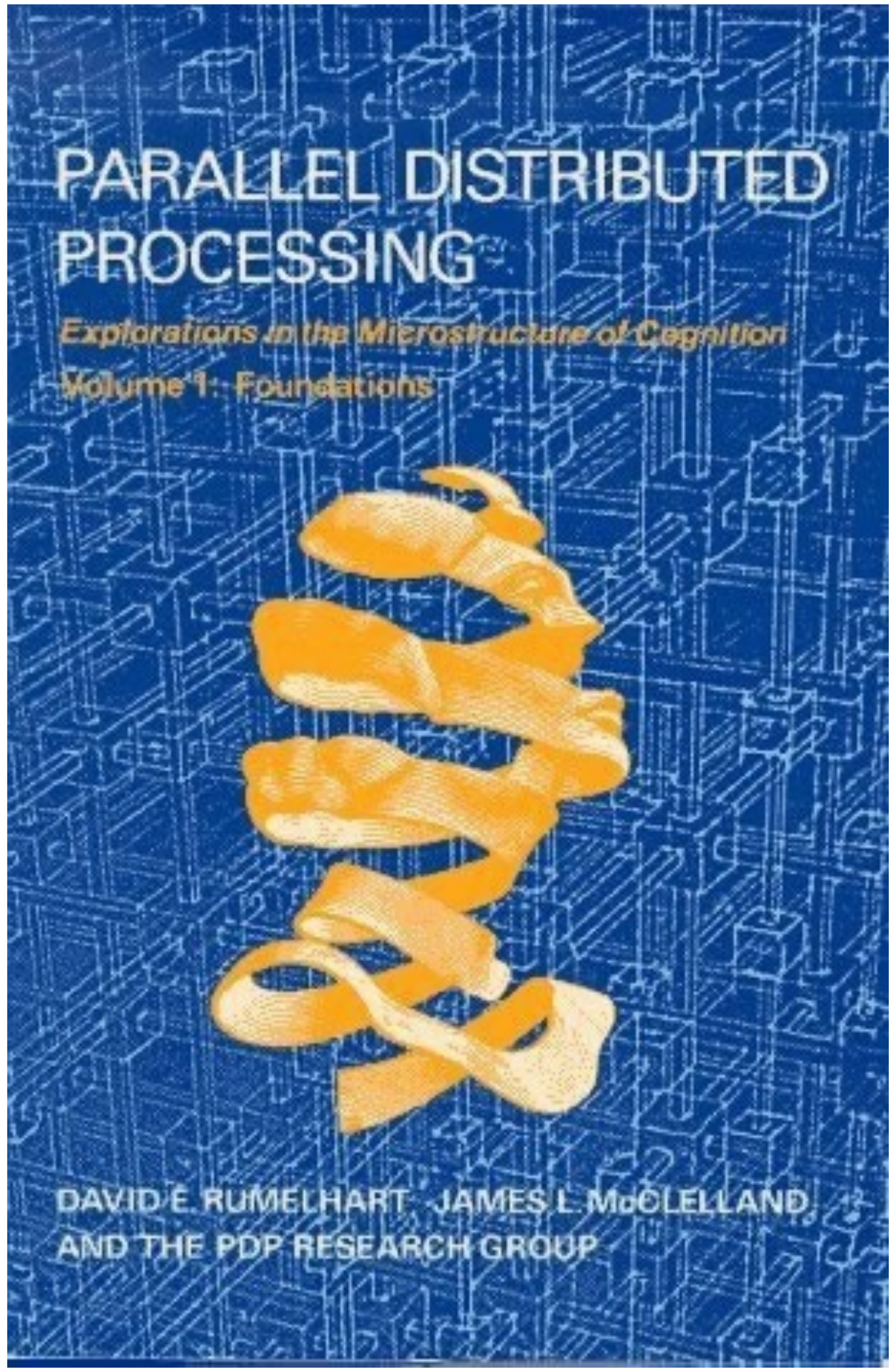
- A **reaction** to the results showing the **limits of Perceptrons**
  - Gave the impression that Perceptrons **can't deliver on promises**
  - **Research funding dried up** (i.e. from the government and other grants)
  - Scientific community **lost interest** in the approach

"AI Winter"

- A **reaction** to the results showing the **limits of Perceptrons**
  - Gave the impression that Perceptrons **can't deliver on promises**
  - **Research funding dried up** (i.e. from the government and other grants)
  - Scientific community **lost interest** in the approach
- The reaction was probably **over-pessimistic**
  - Already known at this time that **any boolean function** can be computed by **deeper networks** of Perceptrons
  - Nonetheless, the technology languished for decades

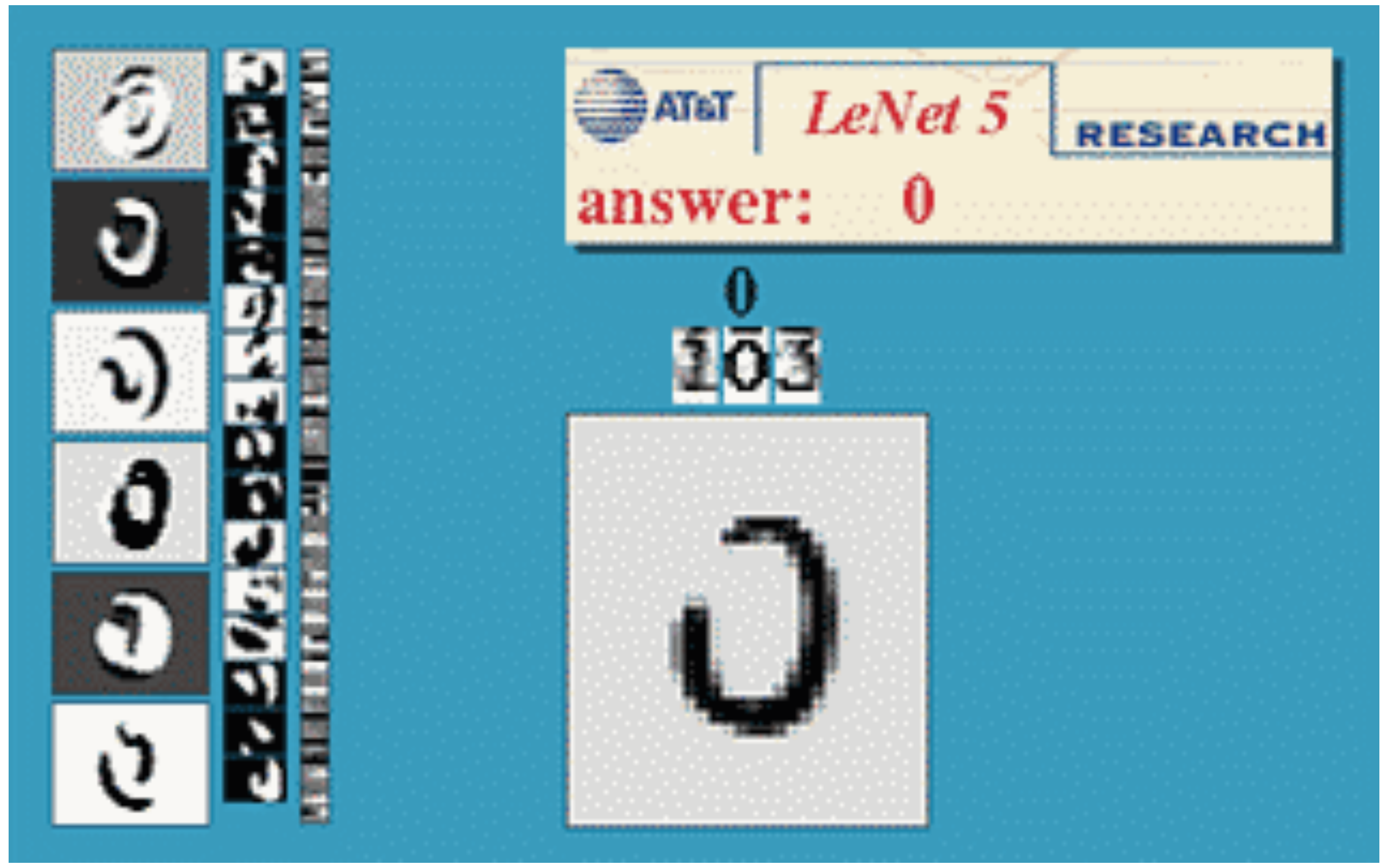
Deeper Backpropagation (1986)

- Presented **multi-layer networks**, trained by the **Backpropagation** algorithm (which we'll discuss)
- "The book *Parallel Distributed Processing* presented the results of some of the first successful experiments with back-propagation in a chapter (Rumelhart et al., 1986b) that contributed greatly to the popularization of back-propagation and initiated a very active period of research in multilayer neural networks."



Successful Engineering Application (1989)

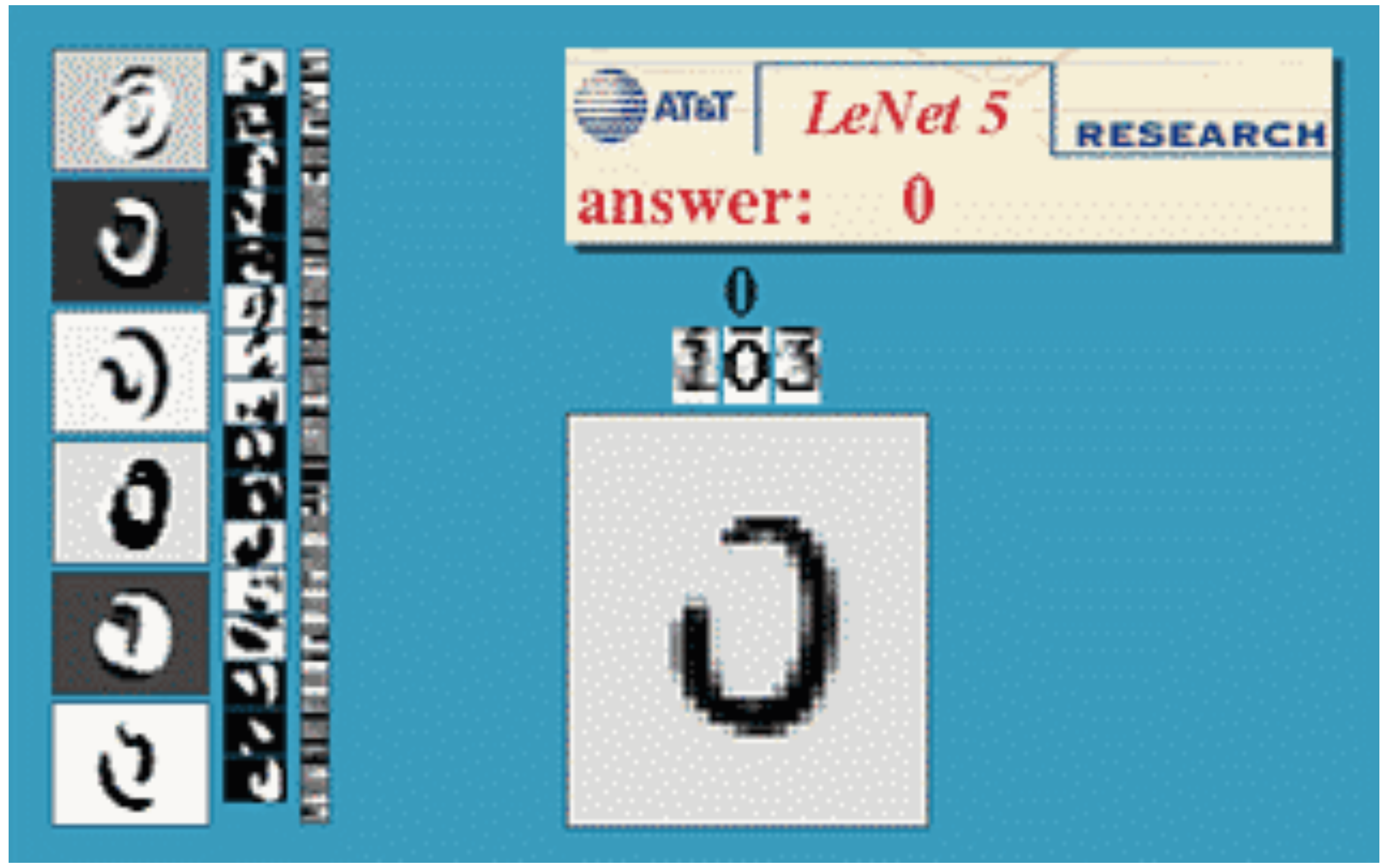
- **Convolutional** networks (“LeNet”, after Yann LeCun) applied to **recognizing hand-written digits**
  - Trained on the MNIST Dataset
  - Long considered the **"Hello World" task** of Deep Learning
- Deployed for automatic reading of **mailing addresses**, **check amounts**, etc.



original website

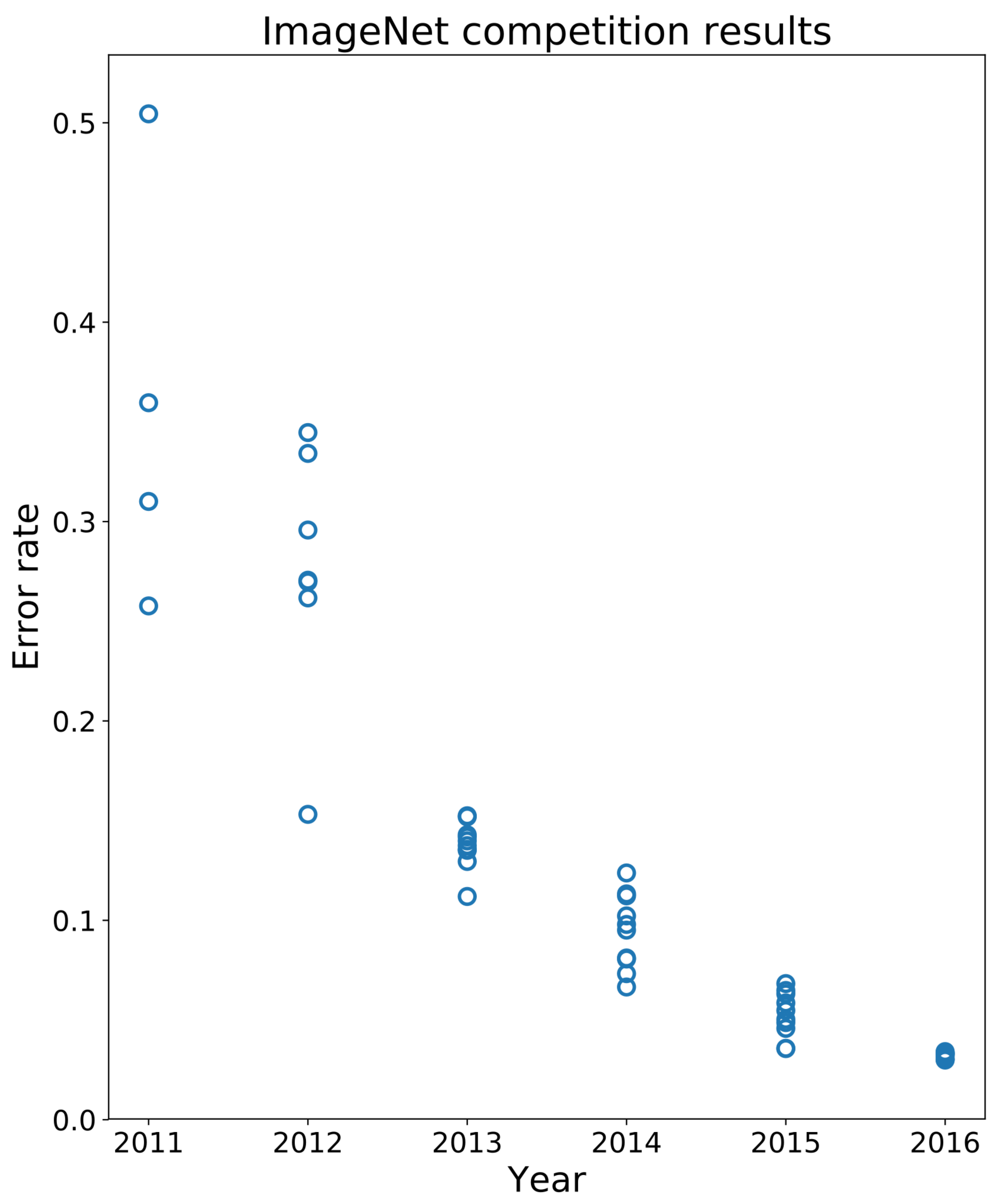
Successful Engineering Application (1989)

- **Convolutional** networks (“LeNet”, after Yann LeCun) applied to **recognizing hand-written digits**
  - Trained on the MNIST Dataset
  - Long considered the **"Hello World" task** of Deep Learning
- Deployed for automatic reading of **mailing addresses**, **check amounts**, etc.



original website

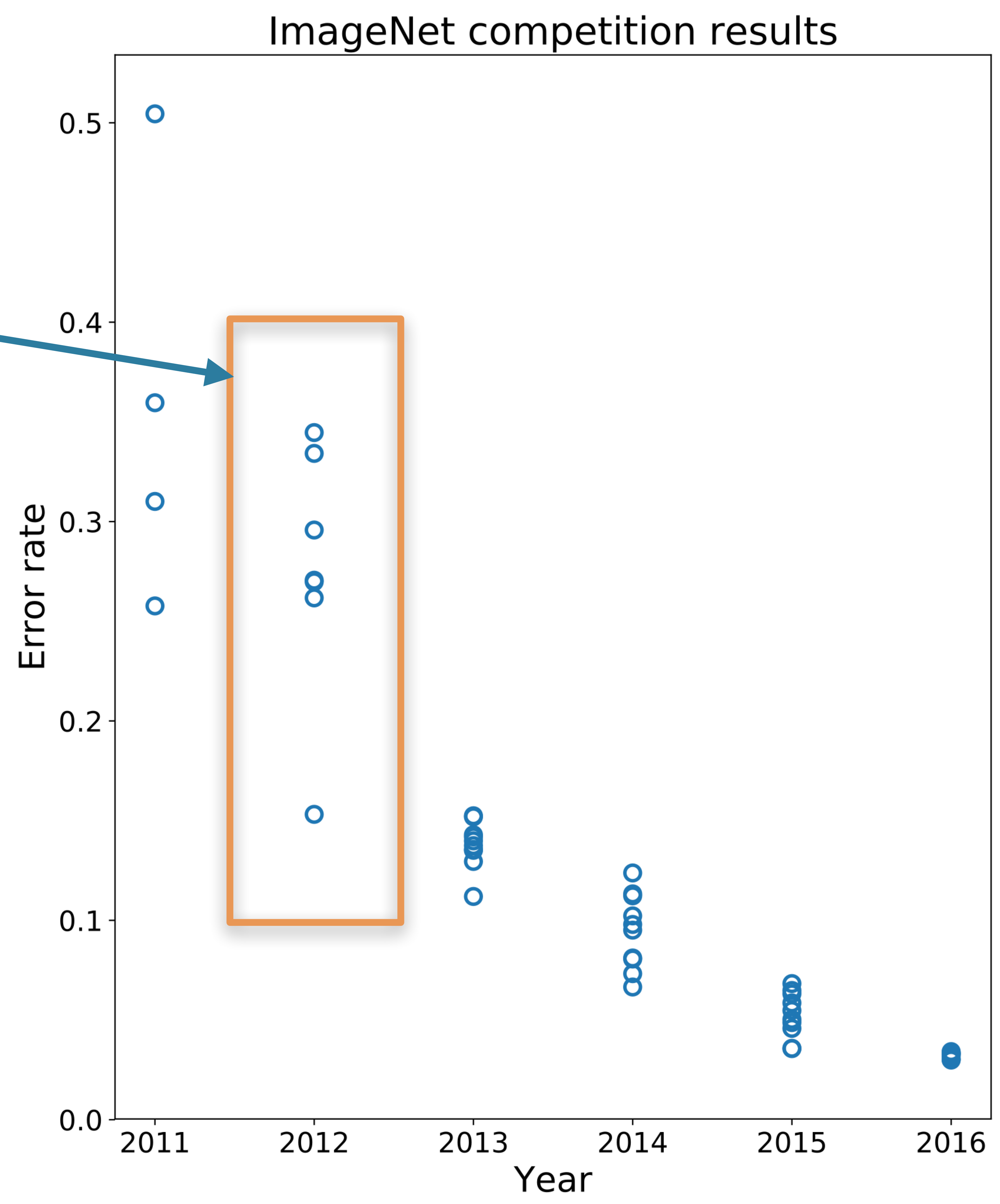
Computer Vision Results (2012)



source

Computer Vision Results (2012)

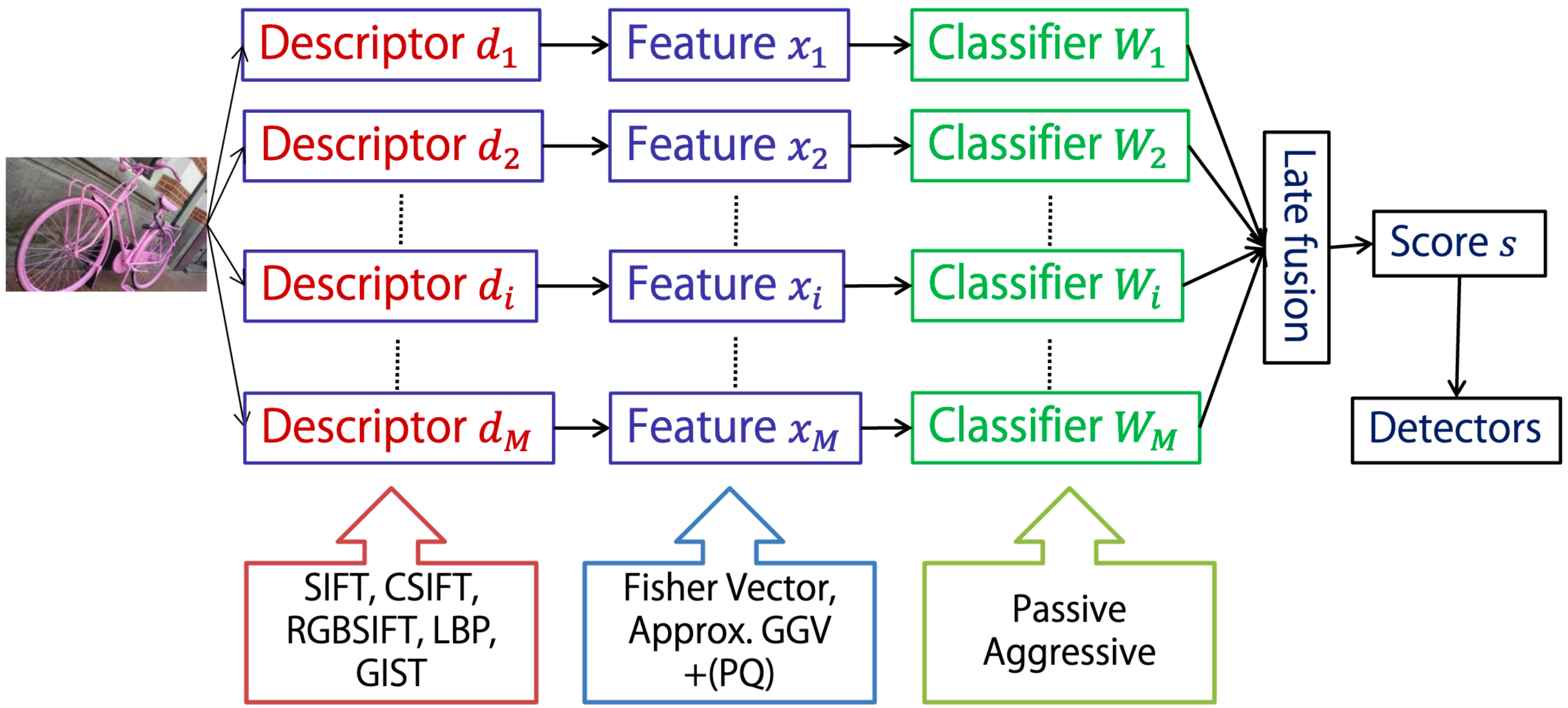
What happened in 2012?



source

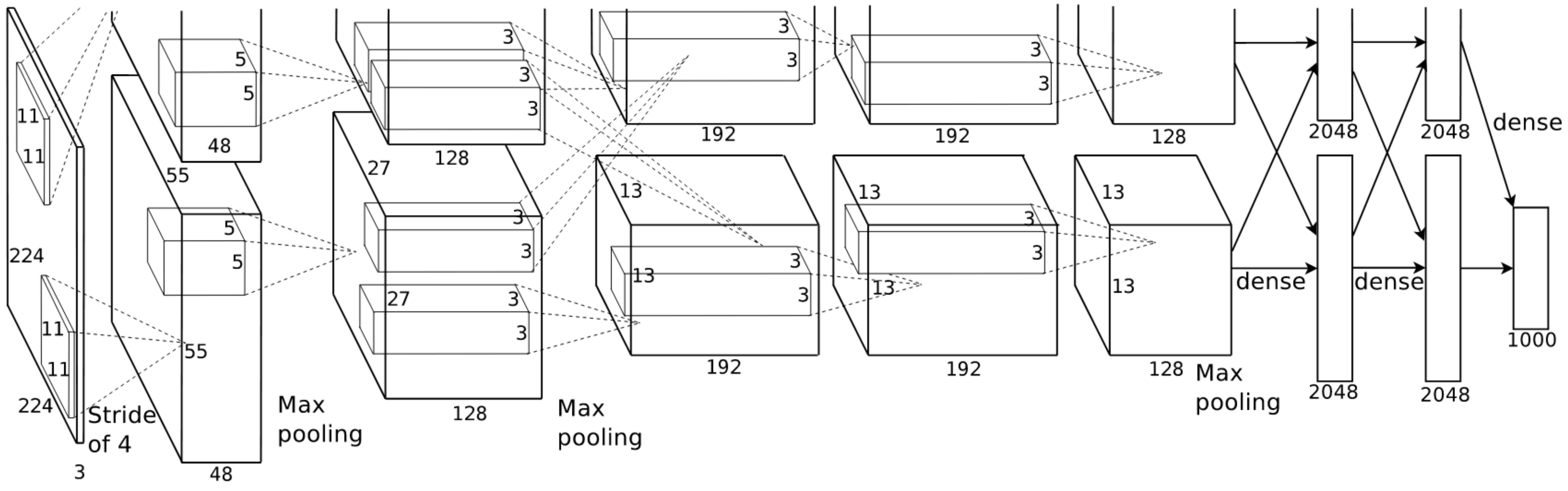
ILSVRC 2012: runner-up

Fisher based features + Multi class linear classifiers



source

ILSVRC 2012: winner



ImageNet Classification with Deep Convolutional Neural Networks

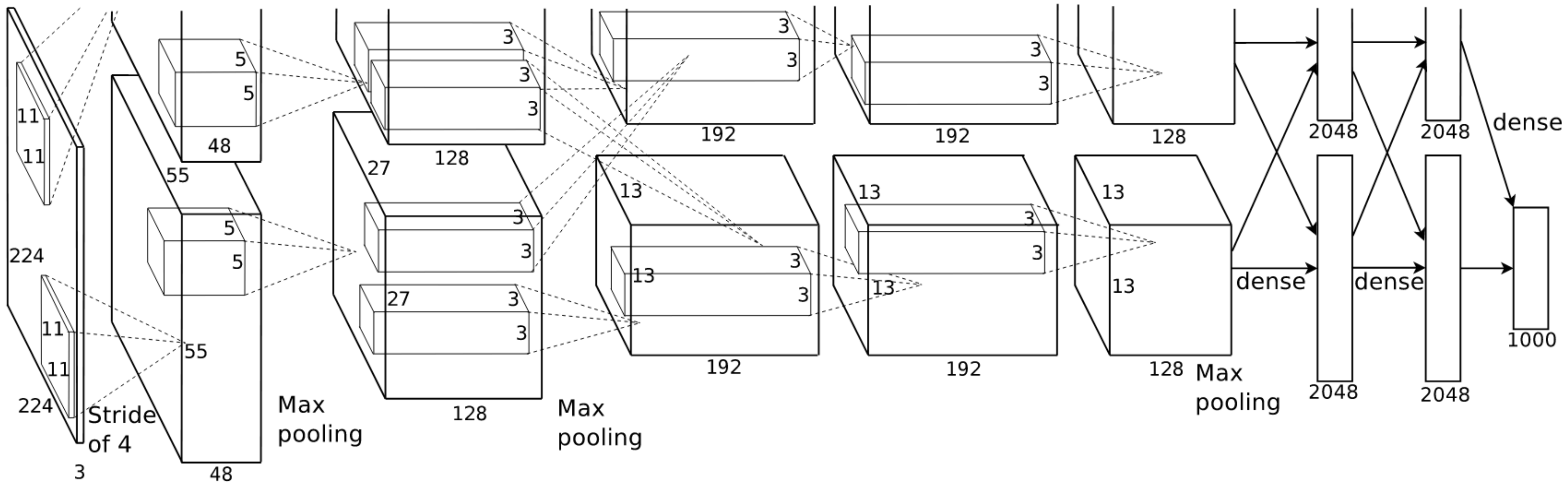
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

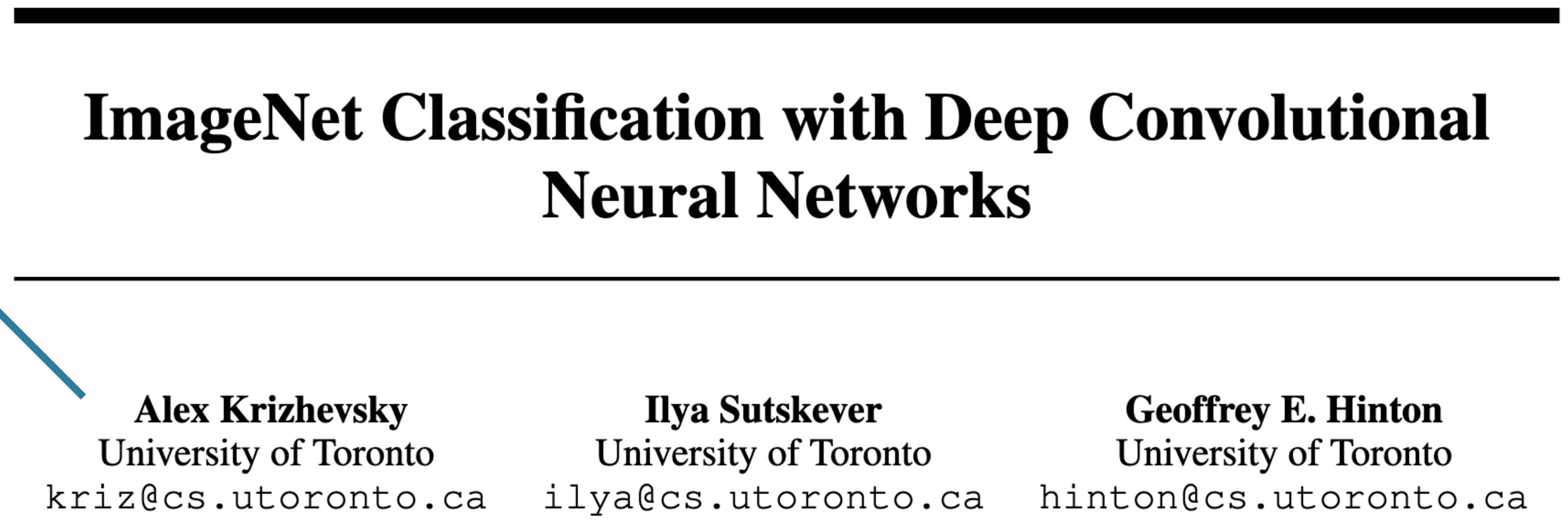
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

NeurIPS 2012 paper

ILSVRC 2012: winner



“AlexNet”



ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

[NeurIPS 2012 paper]

2012-now

2012-now

- Widespread **adoption of deep Neural Networks** across a range of domains
  - **Image processing** / Computer Vision
  - **Game playing** with Reinforcement Learning (e.g. AlphaGo/AlphaZero, …)
  - **Natural Language Processing** (and now LLMs)

2012-now

- Widespread **adoption of deep Neural Networks** across a range of domains
  - **Image processing** / Computer Vision
  - **Game playing** with Reinforcement Learning (e.g. AlphaGo/AlphaZero, …)
  - **Natural Language Processing** (and now LLMs)
- What happened?
  - Better **learning algorithms / training regimes**
  - Larger **standardized datasets**
  - Specialized **computational hardware**
  - Videogames?

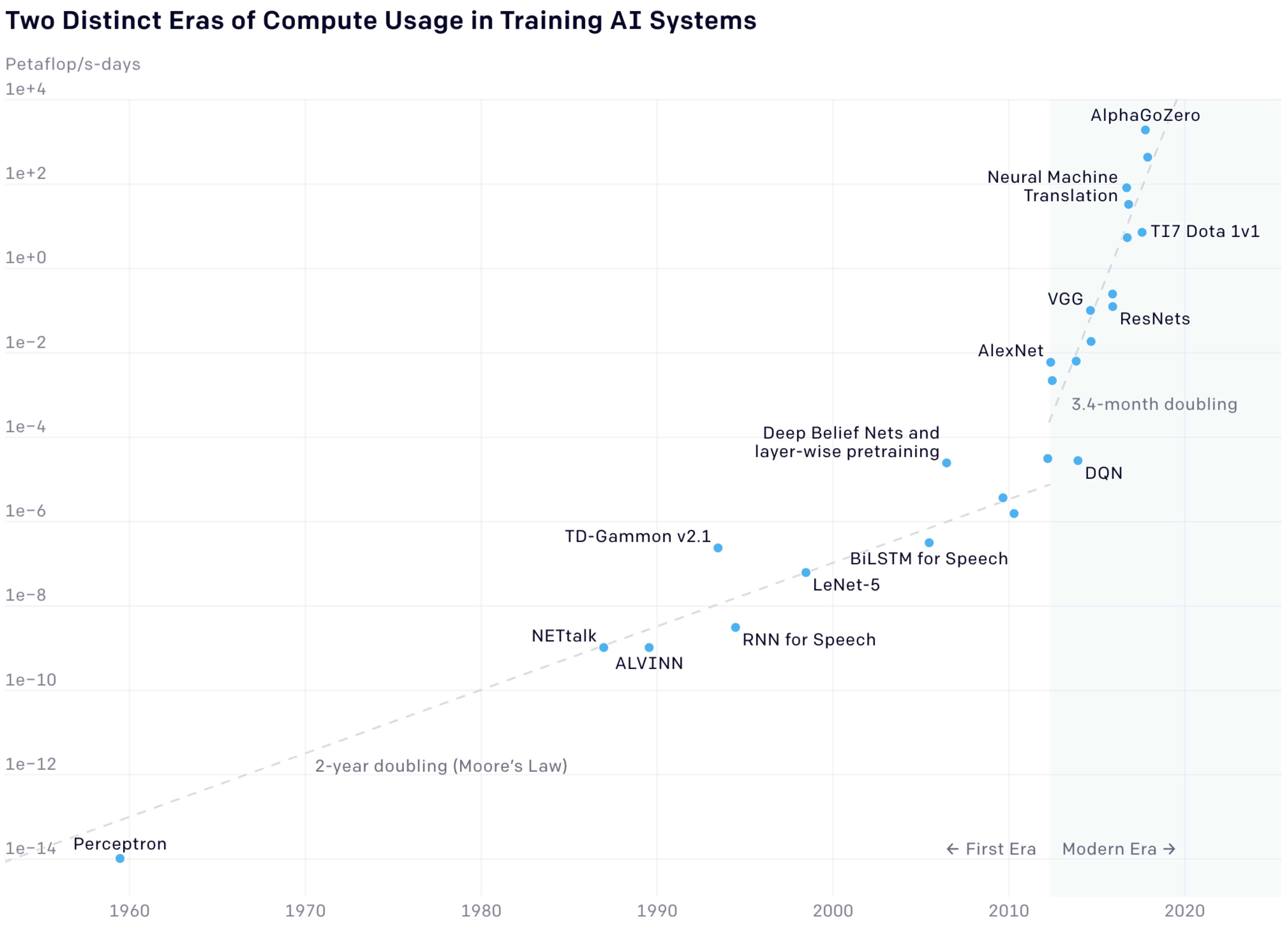
Videogames and Neural Nets

- As it turns out, both 3D graphics and neural networks involve lots of **matrix multiplications**
- The demand for better gaming graphics drove better **Graphics Processing Units (GPUs)**
- The Deep Learning "Revolution" was partially driven by this progress in hardware

Videogames and Neural Nets

- As it turns out, both 3D graphics and neural networks involve lots of **matrix multiplications**
- The demand for better gaming graphics drove better **Graphics Processing Units (GPUs)**
- The Deep Learning "Revolution" was partially driven by this progress in hardware

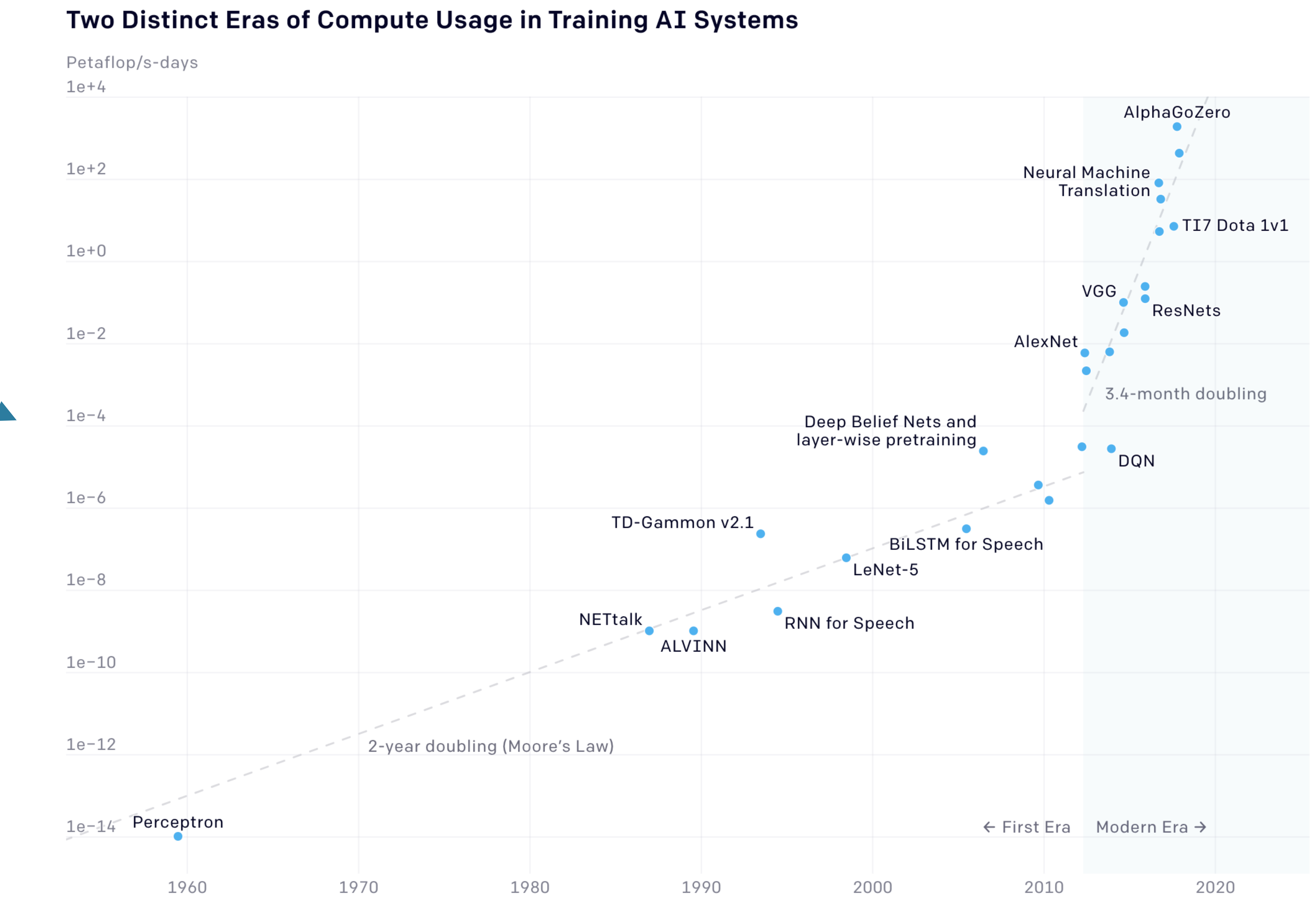
Compute in Deep Learning



source

Compute in Deep Learning

log-scale!!



source

Problems

Problems

- Some areas are an ‘arms race’ between e.g. OpenAI, Meta, Google, MS, Baidu, …

Problems

- Some areas are an ‘arms race’ between e.g. OpenAI, Meta, Google, MS, Baidu, …
- Hugely expensive
  - Carbon emissions
  - Monetarily
    - Inequitable access

Problems

- Some areas are an ‘arms race’ between e.g. Baidu, …
- Hugely expensive
  - Carbon emissions
  - Monetarily
    - Inequitable access

Energy and Policy Considerations for Deep Learning in NLP

Emma Strubell **Ananya Ganesh** **Andrew McCallum**
College of Information and Computer Sciences
University of Massachusetts Amherst
{strubell, aganesh, mccallum}@cs.umass.edu

Abstract

Recent progress in hardware and methodology for training neural networks has ushered in a new generation of large networks trained on abundant data. These models have obtained notable gains in accuracy across many NLP tasks. However, these accuracy improvements depend on the availability of exceptionally large computational resources that necessitate similarly substantial energy consumption. As a result these models are costly to train and develop, both financially, due to the cost of hardware and electricity or cloud compute time, and environmentally, due to the carbon footprint required to fuel modern tensor

Consumption	**CO_2e (lbs)**
Air travel, 1 person, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experiments	78,468
Transformer (big)	192
w/ neural arch. search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Problems

- Some areas are an 'arms race' between e.g. Baidu, …
- Hugely expensive
  - Carbon emissions
  - Monetarily
    - Inequitable access

Energy and Policy Considerations for Deep Learning in NLP

Emma Strubell **Ananya Ganesh** **Andrew McCallum**
College of Information and Computer Sciences
University of Massachusetts Amherst
{strubell, aganesh, mccallum}@cs.umass.edu

Consumption	CO_2e (lbs)
Air travel, 1 person, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experiments	78,468
Transformer (big)	192
w/ neural arch. search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Green AI

Roy Schwartz[*◊] Jesse Dodge[*◊♣] Noah A. Smith[◊♡] Oren Etzioni[◊]

[◊]Allen Institute for AI, Seattle, Washington, USA
[♣] Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[♡] University of Washington, Seattle, Washington, USA

July 2019

Abstract

The computations required for deep learning research have been doubling every few months, resulting in an estimated 300,000x increase from 2012 to 2018 [2]. These computations have a surprisingly large carbon footprint [40]. Ironically, deep learning was inspired by the human brain, which is remarkably energy efficient. Moreover, the financial cost of the computations can make it difficult for academics, students, and researchers, in particular those from emerging economies, to engage in deep learning research.

This position paper advocates a practical solution by making **efficiency** an evaluation criterion for research alongside accuracy and related measures. In addition, we propose reporting the financial cost or "price tag" of developing, training, and running models to provide baselines for the investigation of increasingly efficient methods. Our goal is to make AI both greener and more inclusive—enabling any inspired undergraduate with a laptop to write high-quality research papers. Green AI is an emerging focus at the Allen Institute for AI.

Eras of NLP Models

Four Broad “Eras”

Four Broad "Eras"

- 100% **rule-based** systems (1960s on)

Four Broad "Eras"

- 100% **rule-based** systems (1960s on)
- **Early Machine Learning** (mid-80s - mid-90s)
  - Decision trees, naive bayes, etc

Four Broad "Eras"

- 100% **rule-based** systems (1960s on)
- **Early Machine Learning** (mid-80s - mid-90s)
  - Decision trees, naive bayes, etc
- **Log-linear** (a.k.a. Logistic Regression) models (mid-90s - mid-2010s)

Four Broad "Eras"

- 100% **rule-based** systems (1960s on)
- **Early Machine Learning** (mid-80s - mid-90s)
  - Decision trees, naive bayes, etc
- **Log-linear** (a.k.a. Logistic Regression) models (mid-90s - mid-2010s)
- **Neural Networks** (2013 - now)

Four Broad “Eras”

- 100% **rule-based** systems (1960s on)
- **Early Machine Learning** (mid-80s - mid-90s)
  - Decision trees, naive bayes, etc
- **Log-linear** (a.k.a. Logistic Regression) models (mid-90s - mid-2010s)
- **Neural Networks** (2013 - now)
- **All** of these are **still used** in applications in every area!
  - They all have different strengths and weaknesses

Early NLP Systems (1960s-1990s)

Early NLP Systems (1960s-1990s)

- 100% rule-based, **hand-written algorithms**

Early NLP Systems (1960s-1990s)

- 100% rule-based, **hand-written algorithms**
- Lots of energy in **ontology development** / knowledge representation

Early NLP Systems (1960s-1990s)

- 100% rule-based, **hand-written algorithms**
- Lots of energy in **ontology development** / knowledge representation
- Exhibit many **core features** of human linguistic competence
  - Compositional generalization
  - Abstract representations of meaning

Early NLP Systems (1960s-1990s)

- 100% rule-based, **hand-written algorithms**
- Lots of energy in **ontology development** / knowledge representation
- Exhibit many **core features** of human linguistic competence
  - Compositional generalization
  - Abstract representations of meaning
- **Fully interpretable**, because humans engineered every part

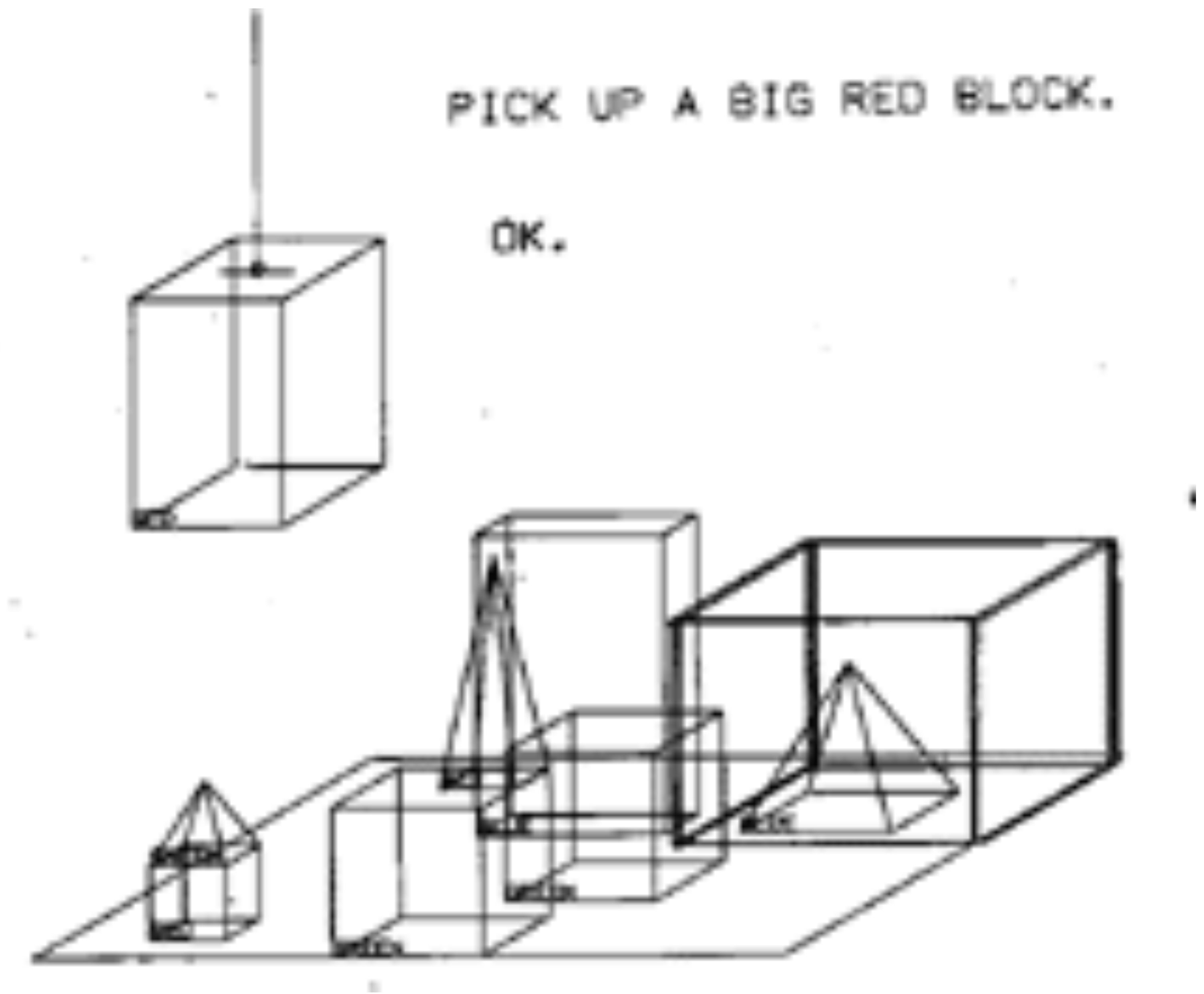
Early NLP Systems (1960s-1990s)

- 100% rule-based, **hand-written algorithms**
- Lots of energy in **ontology development** / knowledge representation
- Exhibit many **core features** of human linguistic competence
  - Compositional generalization
  - Abstract representations of meaning
- **Fully interpretable**, because humans engineered every part
- BUT: **brittle**, no graceful degradation, domain-specific

Early NLP Systems (1960s-1990s)

- SHRDLU, e.g.

Person: Pick up a big red block.
Computer: OK.
Person: Grasp the pyramid.
Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person (changing their mind): Find a block which is taller than the one you are holding and put it into the box.
Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.
Computer: OK.
Person: What does the box contain?
Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.
Person: What is the pyramid supported by?
Computer: THE BOX.



Early ML (80s-90s)

Early ML (80s-90s)

- Spurred by increase in **compute power**, plus availability of **more data**

Early ML (80s-90s)

- Spurred by increase in **compute power**, plus availability of **more data**
- Generally, **Generative models** (models that learn to **generate data**)
  - N-grams, Naive Bayes, HMMs, PCFGs, …

Early ML (80s-90s)

- Spurred by increase in **compute power**, plus availability of **more data**
- Generally, **Generative models** (models that learn to **generate data**)
  - N-grams, Naive Bayes, HMMs, PCFGs, …
- Much of the "learning" simply involved **counting features in a dataset**

Early ML (80s-90s)

- Spurred by increase in **compute power**, plus availability of **more data**
- Generally, **Generative models** (models that learn to **generate data**)
  - N-grams, Naive Bayes, HMMs, PCFGs, …
- Much of the "learning" simply involved **counting features in a dataset**
- Generally relies on heavy use of **feature engineering**
  - I.e. encoding **useful information about the data** by hand / rule

Early ML (80s-90s)

- Spurred by increase in **compute power**, plus availability of **more data**
- Generally, **Generative models** (models that learn to **generate data**)
  - N-grams, Naive Bayes, HMMs, PCFGs, …
- Much of the "learning" simply involved **counting features in a dataset**
- Generally relies on heavy use of **feature engineering**
  - I.e. encoding **useful information about the data** by hand / rule
- These are still **solid baselines** in many tasks

Log-linear Models

- A.K.A. Maximum Entropy (MaxEnt), Logistic Regression, Multinomial Classification
- **Discriminative** models (discriminate the class y based on data x; **do not** try to generate the data)

$$P(y \mid x) \propto e^{\sum_j w_j f_j(x,y)}$$

Log-linear Models

	O	LOC	MISC	ORG	PER
WORDS					
PWORD:at	-0.18	0.94	-0.31	0.28	-0.73
CWORD:Grace	-0.01	0	0	-0.02	0.03
NWORD:Road	0.02	0.27	-0.01	-0.25	-0.03
PWORD-CWORD:at-Grace	0	0	0	0	1 0
CWORD-NWORD:Grace-Road	0	0	0	0	0
NGRAMS (prefix/suffix only here)					
⟨G	-0.57	-0.04	0.26	-0.04	0.45
⟨Gr	0.27	-0.06	0.12	-0.17	-0.16
⟨Gra	-0.01	-0.37	0.19	-0.09	0.28
⟨Grac	-0.01	0	0	-0.02	0.03
⟨Grace	-0.01	0	0	-0.02	0.03
⟨Grace⟩	-0.01	0	0	-0.02	0.03
Grace⟩	-0.01	0	0	-0.02	0.03
race⟩	0	0	0	-0.02	0.03
ace⟩	0.08	0.24	0.07	-0.30	-0.10
ce⟩	0.44	0.31	-0.34	-0.02	-0.38
e⟩	0.38	-0.14	-0.18	-0.06	0
TAGS					
PTAG:IN	-0.40	0.24	0.16	0.08	-0.08
CTAG:NNP	-1.09	0.45	-0.26	0.43	0.47
NTAG:NNP	0.05	-0.19	0.18	-0.12	0.08
PTAG-CTAG:IN-NNP	0	0.14	-0.03	-0.01	-0.10
CTAG-NTAG:NNP-NNP	-0.11	-0.05	0	-0.38	-0.54
TYPES					
PTYPE:x:2	-0.07	-0.15	0.35	0.18	-0.31
CTYPE:Xx	-2.02	0.46	0.19	0.57	0.80
NTYPE:Xx	-0.22	-0.42	-0.19	0.29	0.54
PTYPE-CTYPE:x:2-Xx	-0.20	0.08	0.10	0.10	-0.09
CTYPE-NTYPE:Xx-Xx	0.55	-0.13	-0.55	-0.13	0.26
PTYPE-CTYPE-NTYPE:x:2-Xx-Xx	0.10	0.37	0.10	0.12	-0.69
WORDS/TYPES					
PWORD-CTYPE:at-Xx	-0.21	0.57	-0.21	0.41	-0.56
CTYPE-NWORD:Xx-Road	-0.01	0.27	-0.01	-0.23	-0.03
STATES					
PSTATE:O	2.91	-0.92	-0.72	-0.58	-0.70
PPSTATE-PSTATE:O-O	1.14	-0.60	-0.08	-0.43	-0.04
WORDS/STATES					
PSTATE-CWORD:O-Grace	-0.01	0	0	-0.02	0.03
TAGS/STATES					
PSTATE-PTAG-CTAG:O-IN-NNP	0.12	0.59	-0.29	-0.28	-0.14
PPSTATE-PPTAG-PSTATE-PTAG-CTAG:O-NN-O-IN-NNP	0.01	-0.03	-0.31	0.31	0.01
TYPES/STATES					
PSTATE-CTYPE:O-Xx	-1.13	0.37	-0.12	0.20	0.68
PSTATE-NTYPE:O-Xx	-0.69	-0.3	0.29	0.39	0.30
PSTATE-PTYPE-CTYPE:O-x:2-Xx	-0.28	0.82	-0.10	-0.26	-0.20
PPSTATE-PPTYPE-PSTATE-PTYPE-CTYPE:O-x-O-x:2-Xx	-0.22	-0.04	-0.04	-0.06	0.22
Total:	-1.40	2.68	-1.74	-0.19	-0.58

Log-linear Models

- Learnable using **standard optimization methods**

	O	LOC	MISC	ORG	PER
WORDS					
PWORD:at	-0.18	0.94	-0.31	0.28	-0.73
CWORD:Grace	-0.01	0	0	-0.02	0.03
NWORD:Road	0.02	0.27	-0.01	-0.25	-0.03
PWORD-CWORD:at-Grace	0	0	0	0	1 0
CWORD-NWORD:Grace-Road	0	0	0	0	0
NGRAMS (prefix/suffix only here)					
⟨G	-0.57	-0.04	0.26	-0.04	0.45
⟨Gr	0.27	-0.06	0.12	-0.17	-0.16
⟨Gra	-0.01	-0.37	0.19	-0.09	0.28
⟨Grac	-0.01	0	0	-0.02	0.03
⟨Grace	-0.01	0	0	-0.02	0.03
⟨Grace⟩	-0.01	0	0	-0.02	0.03
Grace⟩	-0.01	0	0	-0.02	0.03
race⟩	0	0	0	-0.02	0.03
ace⟩	0.08	0.24	0.07	-0.30	-0.10
ce⟩	0.44	0.31	-0.34	-0.02	-0.38
e⟩	0.38	-0.14	-0.18	-0.06	0
TAGS					
PTAG:IN	-0.40	0.24	0.16	0.08	-0.08
CTAG:NNP	-1.09	0.45	-0.26	0.43	0.47
NTAG:NNP	0.05	-0.19	0.18	-0.12	0.08
PTAG-CTAG:IN-NNP	0	0.14	-0.03	-0.01	-0.10
CTAG-NTAG:NNP-NNP	-0.11	-0.05	0	-0.38	-0.54
TYPES					
PTYPE:x:2	-0.07	-0.15	0.35	0.18	-0.31
CTYPE:Xx	-2.02	0.46	0.19	0.57	0.80
NTYPE:Xx	-0.22	-0.42	-0.19	0.29	0.54
PTYPE-CTYPE:x:2-Xx	-0.20	0.08	0.10	0.10	-0.09
CTYPE-NTYPE:Xx-Xx	0.55	-0.13	-0.55	-0.13	0.26
PTYPE-CTYPE-NTYPE:x:2-Xx-Xx	0.10	0.37	0.10	0.12	-0.69
WORDS/TYPES					
PWORD-CTYPE:at-Xx	-0.21	0.57	-0.21	0.41	-0.56
CTYPE-NWORD:Xx-Road	-0.01	0.27	-0.01	-0.23	-0.03
STATES					
PSTATE:O	2.91	-0.92	-0.72	-0.58	-0.70
PPSTATE-PSTATE:O-O	1.14	-0.60	-0.08	-0.43	-0.04
WORDS/STATES					
PSTATE-CWORD:O-Grace	-0.01	0	0	-0.02	0.03
TAGS/STATES					
PSTATE-PTAG-CTAG:O-IN-NNP	0.12	0.59	-0.29	-0.28	-0.14
PPSTATE-PPTAG-PSTATE-PTAG-CTAG:O-NN-O-IN-NNP	0.01	-0.03	-0.31	0.31	0.01
TYPES/STATES					
PSTATE-CTYPE:O-Xx	-1.13	0.37	-0.12	0.20	0.68
PSTATE-NTYPE:O-Xx	-0.69	-0.3	0.29	0.39	0.30
PSTATE-PTYPE-CTYPE:O-x:2-Xx	-0.28	0.82	-0.10	-0.26	-0.20
PPSTATE-PPTYPE-PSTATE-PTYPE-CTYPE:O-x-O-x:2-Xx	-0.22	-0.04	-0.04	-0.06	0.22
Total:	-1.40	2.68	-1.74	-0.19	-0.58

Log-linear Models

- Learnable using **standard optimization methods**
- **Interpretable**: can see which features are important
  - e.g. Klein et al 2003 on Named Entity Recognition:
    - Weight for class PER for feature CURWORD:Grace: 0.03
    - Weight for class PER for prefix “<G”: 0.45

	O	LOC	MISC	ORG	PER
WORDS					
PWORD:at	-0.18	0.94	-0.31	0.28	-0.73
CWORD:Grace	-0.01	0	0	-0.02	0.03
NWORD:Road	0.02	0.27	-0.01	-0.25	-0.03
PWORD-CWORD:at-Grace	0	0	0	0	1 0
CWORD-NWORD:Grace-Road	0	0	0	0	0
NGRAMS (prefix/suffix only here)					
⟨G	-0.57	-0.04	0.26	-0.04	0.45
⟨Gr	0.27	-0.06	0.12	-0.17	-0.16
⟨Gra	-0.01	-0.37	0.19	-0.09	0.28
⟨Grac	-0.01	0	0	-0.02	0.03
⟨Grace	-0.01	0	0	-0.02	0.03
⟨Grace⟩	-0.01	0	0	-0.02	0.03
Grace⟩	-0.01	0	0	-0.02	0.03
race⟩	0	0	0	-0.02	0.03
ace⟩	0.08	0.24	0.07	-0.30	-0.10
ce⟩	0.44	0.31	-0.34	-0.02	-0.38
e⟩	0.38	-0.14	-0.18	-0.06	0
TAGS					
PTAG:IN	-0.40	0.24	0.16	0.08	-0.08
CTAG:NNP	-1.09	0.45	-0.26	0.43	0.47
NTAG:NNP	0.05	-0.19	0.18	-0.12	0.08
PTAG-CTAG:IN-NNP	0	0.14	-0.03	-0.01	-0.10
CTAG-NTAG:NNP-NNP	-0.11	-0.05	0	-0.38	-0.54
TYPES					
PTYPE:x:2	-0.07	-0.15	0.35	0.18	-0.31
CTYPE:Xx	-2.02	0.46	0.19	0.57	0.80
NTYPE:Xx	-0.22	-0.42	-0.19	0.29	0.54
PTYPE-CTYPE:x:2-Xx	-0.20	0.08	0.10	0.10	-0.09
CTYPE-NTYPE:Xx-Xx	0.55	-0.13	-0.55	-0.13	0.26
PTYPE-CTYPE-NTYPE:x:2-Xx-Xx	0.10	0.37	0.10	0.12	-0.69
WORDS/TYPES					
PWORD-CTYPE:at-Xx	-0.21	0.57	-0.21	0.41	-0.56
CTYPE-NWORD:Xx-Road	-0.01	0.27	-0.01	-0.23	-0.03
STATES					
PSTATE:O	2.91	-0.92	-0.72	-0.58	-0.70
PPSTATE-PSTATE:O-O	1.14	-0.60	-0.08	-0.43	-0.04
WORDS/STATES					
PSTATE-CWORD:O-Grace	-0.01	0	0	-0.02	0.03
TAGS/STATES					
PSTATE-PTAG-CTAG:O-IN-NNP	0.12	0.59	-0.29	-0.28	-0.14
PPSTATE-PPTAG-PSTATE-PTAG-CTAG:O-NN-O-IN-NNP	0.01	-0.03	-0.31	0.31	0.01
TYPES/STATES					
PSTATE-CTYPE:O-Xx	-1.13	0.37	-0.12	0.20	0.68
PSTATE-NTYPE:O-Xx	-0.69	-0.3	0.29	0.39	0.30
PSTATE-PTYPE-CTYPE:O-x:2-Xx	-0.28	0.82	-0.10	-0.26	-0.20
PPSTATE-PPTYPE-PSTATE-PTYPE-CTYPE:O-x-O-x:2-Xx	-0.22	-0.04	-0.04	-0.06	0.22
Total:	-1.40	2.68	-1.74	-0.19	-0.58

Log-linear Models

- Learnable using **standard optimization methods**
- **Interpretable**: can see which features are important
  - e.g. Klein et al 2003 on Named Entity Recognition:
    - Weight for class PER for feature CURWORD:Grace: 0.03
    - Weight for class PER for prefix “<G”: 0.45
- Feature engineering is...
  - **Expensive** (takes and expert much time to annotate)
  - **Incomplete** (might be useful features the expert misses)
  - **Sparse** (some features **aren't useful** so the compute is wasted

	O	LOC	MISC	ORG	PER
WORDS					
PWORD:at	-0.18	0.94	-0.31	0.28	-0.73
CWORD:Grace	-0.01	0	0	-0.02	0.03
NWORD:Road	0.02	0.27	-0.01	-0.25	-0.03
PWORD-CWORD:at-Grace	0	0	0	0	1 0
CWORD-NWORD:Grace-Road	0	0	0	0	0
NGRAMS (prefix/suffix only here)					
⟨G	-0.57	-0.04	0.26	-0.04	0.45
⟨Gr	0.27	-0.06	0.12	-0.17	-0.16
⟨Gra	-0.01	-0.37	0.19	-0.09	0.28
⟨Grac	-0.01	0	0	-0.02	0.03
⟨Grace	-0.01	0	0	-0.02	0.03
⟨Grace⟩	-0.01	0	0	-0.02	0.03
Grace⟩	-0.01	0	0	-0.02	0.03
race⟩	0	0	0	-0.02	0.03
ace⟩	0.08	0.24	0.07	-0.30	-0.10
ce⟩	0.44	0.31	-0.34	-0.02	-0.38
e⟩	0.38	-0.14	-0.18	-0.06	0
TAGS					
PTAG:IN	-0.40	0.24	0.16	0.08	-0.08
CTAG:NNP	-1.09	0.45	-0.26	0.43	0.47
NTAG:NNP	0.05	-0.19	0.18	-0.12	0.08
PTAG-CTAG:IN-NNP	0	0.14	-0.03	-0.01	-0.10
CTAG-NTAG:NNP-NNP	-0.11	-0.05	0	-0.38	-0.54
TYPES					
PTYPE:x:2	-0.07	-0.15	0.35	0.18	-0.31
CTYPE:Xx	-2.02	0.46	0.19	0.57	0.80
NTYPE:Xx	-0.22	-0.42	-0.19	0.29	0.54
PTYPE-CTYPE:x:2-Xx	-0.20	0.08	0.10	0.10	-0.09
CTYPE-NTYPE:Xx-Xx	0.55	-0.13	-0.55	-0.13	0.26
PTYPE-CTYPE-NTYPE:x:2-Xx-Xx	0.10	0.37	0.10	0.12	-0.69
WORDS/TYPES					
PWORD-CTYPE:at-Xx	-0.21	0.57	-0.21	0.41	-0.56
CTYPE-NWORD:Xx-Road	-0.01	0.27	-0.01	-0.23	-0.03
STATES					
PSTATE:O	2.91	-0.92	-0.72	-0.58	-0.70
PPSTATE-PSTATE:O-O	1.14	-0.60	-0.08	-0.43	-0.04
WORDS/STATES					
PSTATE-CWORD:O-Grace	-0.01	0	0	-0.02	0.03
TAGS/STATES					
PSTATE-PTAG-CTAG:O-IN-NNP	0.12	0.59	-0.29	-0.28	-0.14
PPSTATE-PPTAG-PSTATE-PTAG-CTAG:O-NN-O-IN-NNP	0.01	-0.03	-0.31	0.31	0.01
TYPES/STATES					
PSTATE-CTYPE:O-Xx	-1.13	0.37	-0.12	0.20	0.68
PSTATE-NTYPE:O-Xx	-0.69	-0.3	0.29	0.39	0.30
PSTATE-PTYPE-CTYPE:O-x:2-Xx	-0.28	0.82	-0.10	-0.26	-0.20
PPSTATE-PPTYPE-PSTATE-PTYPE-CTYPE:O-x-O-x:2-Xx	-0.22	-0.04	-0.04	-0.06	0.22
Total:	-1.40	2.68	-1.74	-0.19	-0.58

Engineered Features

	O	LOC	MISC	ORG	PER
WORDS					
PWORD:at	-0.18	0.94	-0.31	0.28	-0.73
CWORD:Grace	-0.01	0	0	-0.02	0.03
NWORD:Road	0.02	0.27	-0.01	-0.25	-0.03
PWORD-CWORD:at-Grace	0	0	0	0	1 0
CWORD-NWORD:Grace-Road	0	0	0	0	0
NGRAMS (prefix/suffix only here)					
⟨G	-0.57	-0.04	0.26	-0.04	0.45
⟨Gr	0.27	-0.06	0.12	-0.17	-0.16
⟨Gra	-0.01	-0.37	0.19	-0.09	0.28
⟨Grac	-0.01	0	0	-0.02	0.03
⟨Grace	-0.01	0	0	-0.02	0.03
⟨Grace⟩	-0.01	0	0	-0.02	0.03
Grace⟩	-0.01	0	0	-0.02	0.03
race⟩	0	0	0	-0.02	0.03
ace⟩	0.08	0.24	0.07	-0.30	-0.10
ce⟩	0.44	0.31	-0.34	-0.02	-0.38
e⟩	0.38	-0.14	-0.18	-0.06	0
TAGS					
PTAG:IN	-0.40	0.24	0.16	0.08	-0.08
CTAG:NNP	-1.09	0.45	-0.26	0.43	0.47
NTAG:NNP	0.05	-0.19	0.18	-0.12	0.08
PTAG-CTAG:IN-NNP	0	0.14	-0.03	-0.01	-0.10
CTAG-NTAG:NNP-NNP	-0.11	-0.05	0	-0.38	-0.54

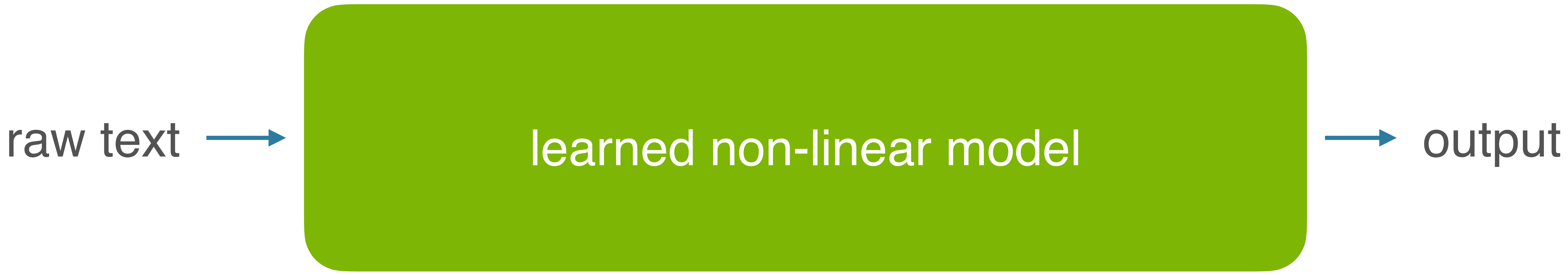
Neural Networks

Neural Networks

- Key idea: **no feature engineering**
  - Have a larger model **learn which features are useful**
  - (but **can be combined with feature extraction** as well)

Neural Networks

- Key idea: **no feature engineering**
  - Have a larger model **learn which features are useful**
  - (but **can be combined with feature extraction** as well)
- **“End-to-end”** learning paradigm:



Neural Networks (drawbacks)

Neural Networks (drawbacks)

- Regarded as **uninterpretable** (**"Black Boxes"**)
  - How do we know **what the model has learned?**
  - Can we **trust** it in deployment? (Just look at headlines for cases where it goes wrong!)
  - Sometimes learns to **solve a dataset** it was trained on, but not generalize

Neural Networks (drawbacks)

- Regarded as **uninterpretable** (**"Black Boxes"**)
  - How do we know **what the model has learned?**
  - Can we **trust** it in deployment? (Just look at headlines for cases where it goes wrong!)
  - Sometimes learns to **solve a dataset** it was trained on, but not generalize
- Large-to-astronomical **computational requirements** (with environmental effects)

Neural Networks (drawbacks)

- Regarded as **uninterpretable** (**"Black Boxes"**)
  - How do we know **what the model has learned?**
  - Can we **trust** it in deployment? (Just look at headlines for cases where it goes wrong!)
  - Sometimes learns to **solve a dataset** it was trained on, but not generalize
- Large-to-astronomical **computational requirements** (with environmental effects)
- Large-to-astronomical **data requirements**, raising issues like:
  - **Documentation debt** (what exactly went in to it?)
  - **Privacy and copyright** concerns
  - Amplifying **biases**

Neural Networks (drawbacks)

- Regarded
  - How do
  - Can we … es wrong!)
  - Sometin
- Large-to- … al effects)
- Large-to-
  - **Docu**
  - **Priva**
  - Ampli…

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

Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

Shmargaret Shmitchell
shmargaret.shmitchell@gmail.com
The Aether

ABSTRACT

The past 3 years of work in NLP have been characterized by the development and deployment of ever larger language models, especially for English. BERT, its variants, GPT-2/3, and others, most recently Switch-C, have pushed the boundaries of the possible both through architectural innovations and through sheer size. Using these pretrained models and the methodology of fine-tuning them for specific tasks, researchers have extended the state of the art on a wide array of tasks as measured by leaderboards on specific benchmarks for English. In this paper, we take a step back and ask:

alone, we have seen the emergence of BERT and its variants [39, 70, 74, 113, 146], GPT-2 [106], T-NLG [112], GPT-3 [25], and most recently Switch-C [43], with institutions seemingly competing to produce ever larger LMs. While investigating properties of LMs and how they change with size holds scientific interest, and large LMs have shown improvements on various tasks (§2), we ask whether enough thought has been put into the potential risks associated with developing them and strategies to mitigate these risks.

We first consider environmental risks. Echoing a line of recent work outlining the environmental and financial costs of deep learn-

Questions?

Next time: Vectors, Matrices, and Linear Transformations