

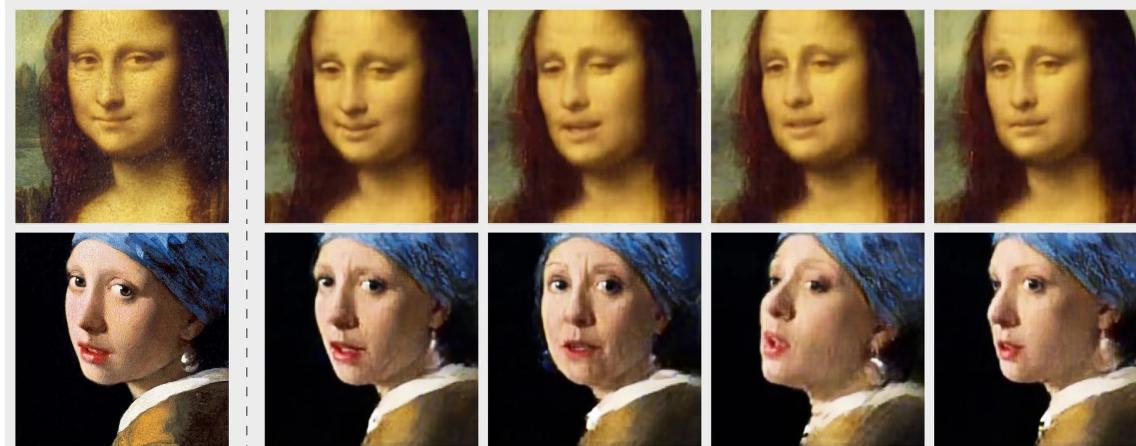
ICME Summer Workshops 2020

Introduction to Deep Learning

Session 1: 9:00 – 10:30 AM

Instructor: Sherrie Wang

icme-workshops.github.io/deep-learning



Zakharov et al. "Few-Shot Adversarial Learning of Realistic Neural Talking Head Models" (2019)

Workshop Schedule

Session 1 (9:00–10:30 AM)

- Introduction
- Current state-of-the-art in deep learning
- Math review
- Fully connected neural networks

Session 2 (10:45–12:00 PM)

- Loss functions
- Gradient descent
- Backpropagation
- Overfitting and underfitting

Lunch (12:00–2:00 PM)

Session 3 (2:00–3:15 PM)

- Convolutional neural networks
- Recurrent neural networks
- Other architectures
- Deep learning libraries
- Hands-on coding session in Tensorflow

Session 4 (3:30–4:45 PM)

- Hands-on coding session in Keras
- Hands-on coding session on transfer learning
- Failures of deep learning

About me



Hello
my name is

Sherrie Wang



@sherwang

- **PhD student** in ICME, Year 5
- **Research focus:** Machine learning methods for remote sensing and applications in sustainability
- **Relevant courses:**
 - CS 221 (Artificial Intelligence)
 - CS 229 (Machine Learning)
 - CS 228 (Probabilistic Graphical Models)
 - CS 230 (Deep Learning)
 - CS 231n (Convolutional Neural Networks)
 - CS 236 (Generative Adversarial Networks)
 - CS 330 (Deep Multi-Task and Meta Learning)
- **Relevant teaching:** CME 250 (Introduction to Machine Learning), Summer Workshop 2019

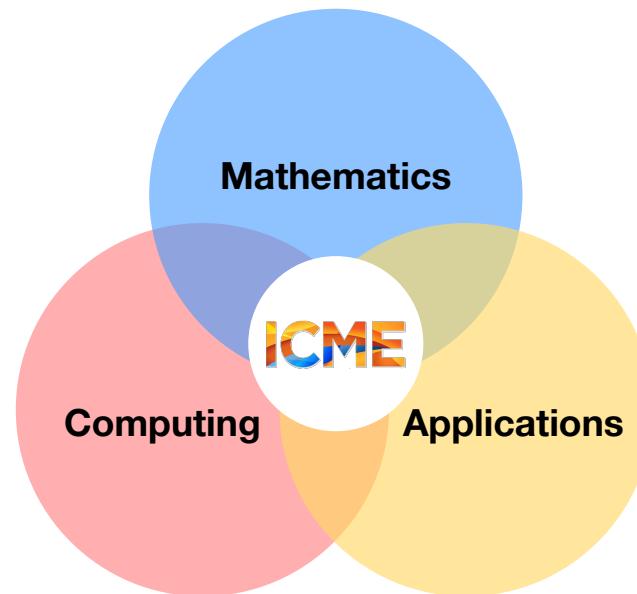
What is **ICME**?

Stanford Institute for Computational and Mathematical Engineering

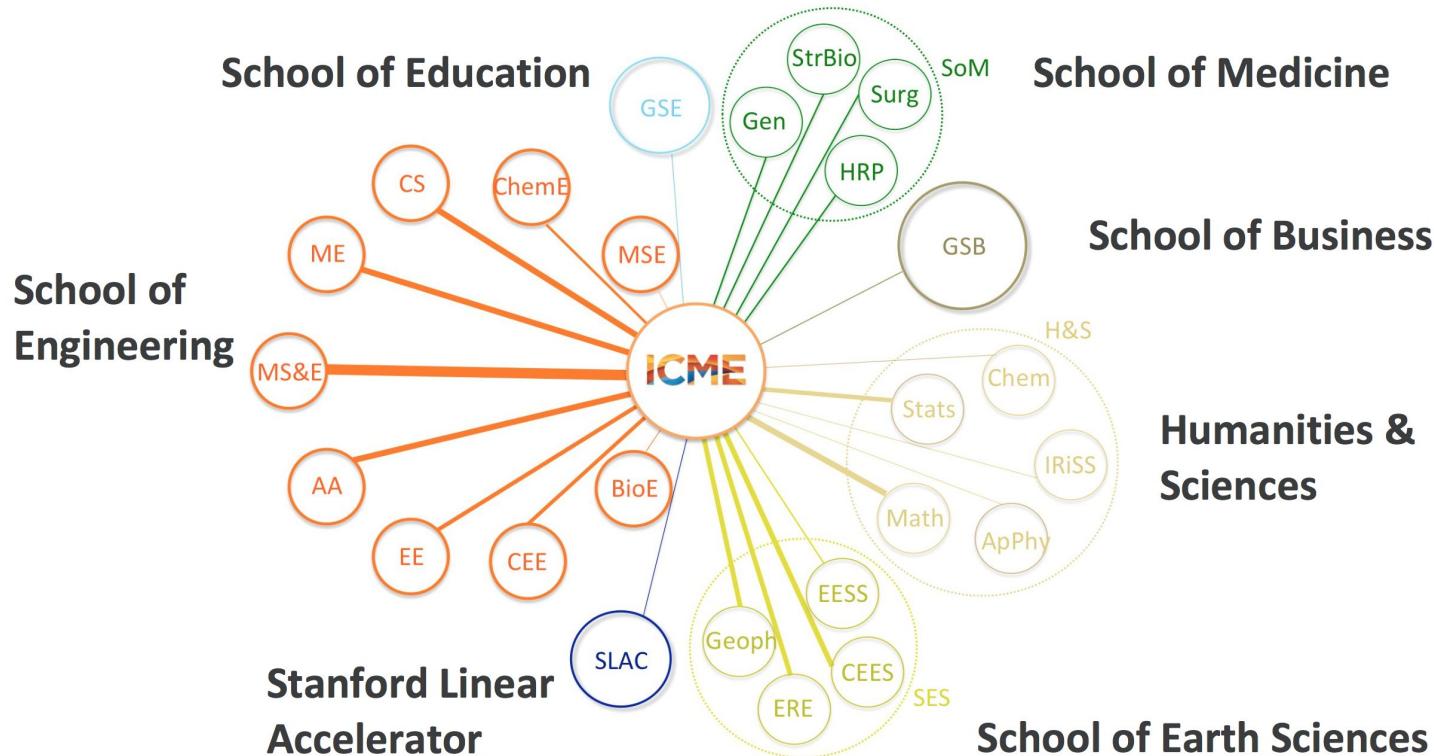
Hub at Stanford

Interdisciplinary field using advanced mathematics and computing to address complex problems

icme.stanford.edu



55+ affiliated faculty across Stanford



150+ MS and PhD students



ICME industry-academia ecosystem



Schlumberger

Tencent 腾讯

Google

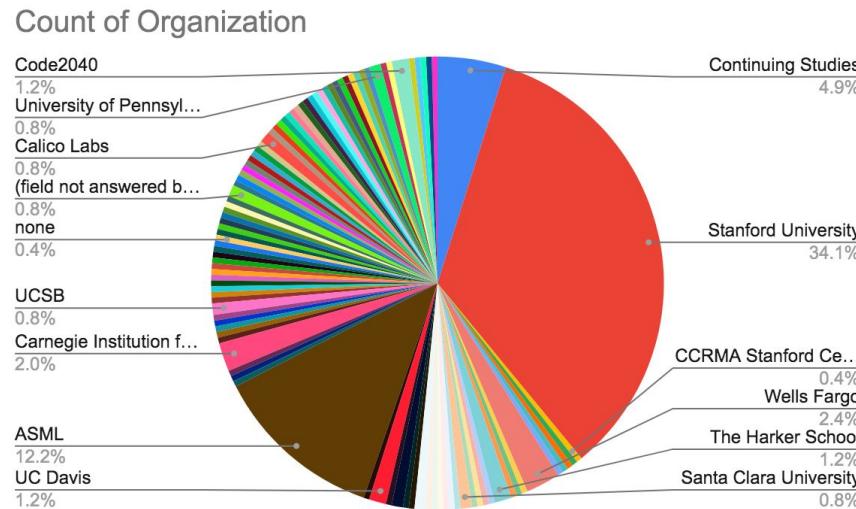
MathWorks

@WalmartLabs

Accenture Applied Intelligence

Who's here today

250+ registered participants



Go to PollEv.com/dlworkshop2020

Job title:

- 34% student
- 17% in research
- 5% continuing studies
- 4% software engineer
- 4% professor

Academia / industry:

- 66% academia
- 34% industry

The Deep Learning Revolution

What is “deep learning”?

Artificial intelligence

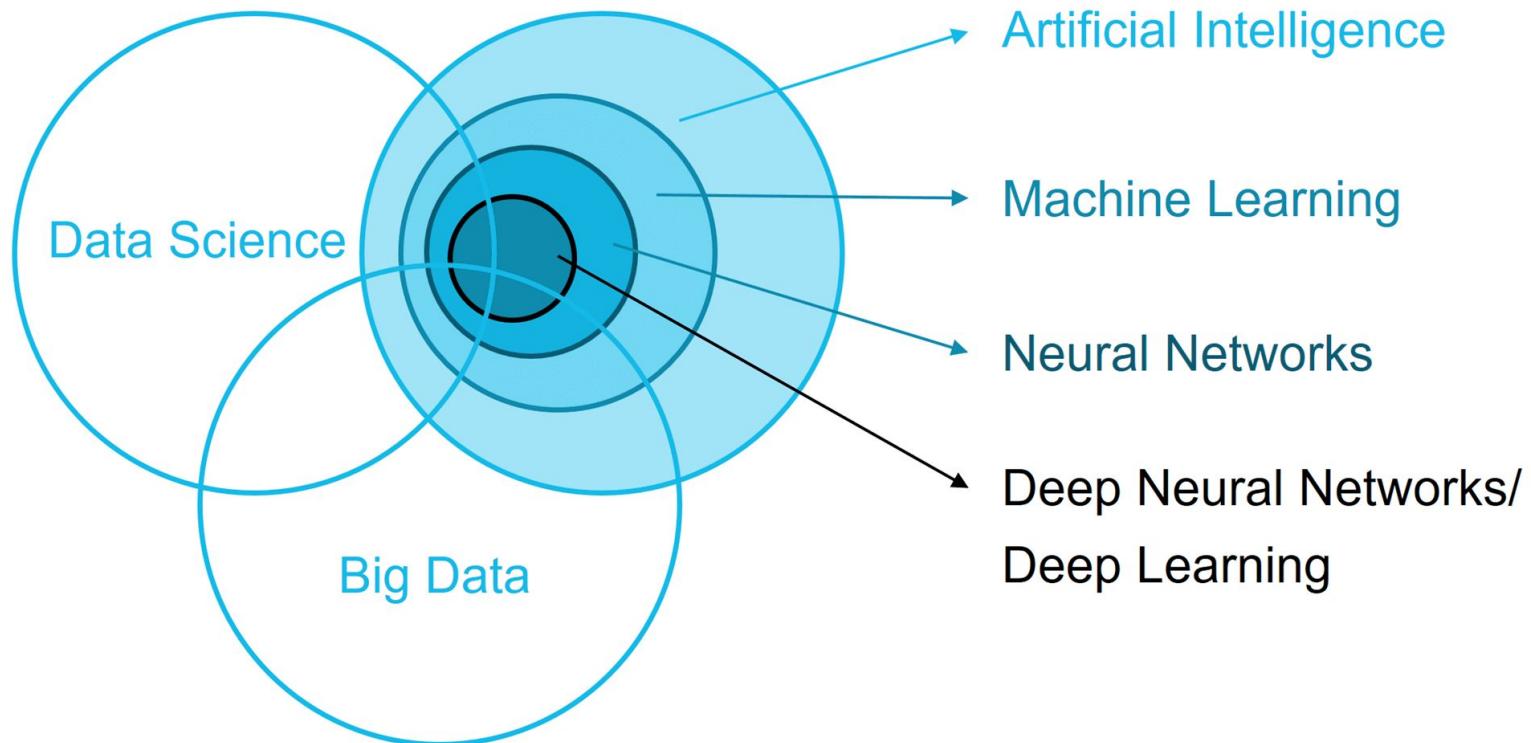
Big data

Machine learning

Deep learning

Data science

What is “deep learning”?

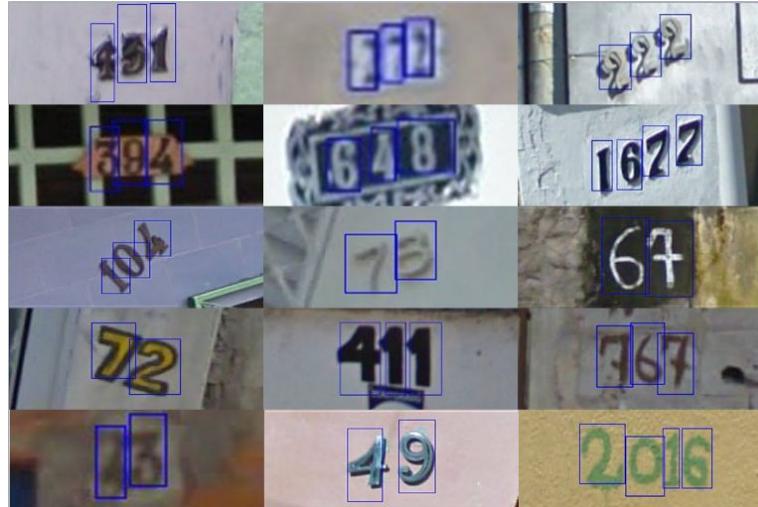


Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

1. Optical character recognition

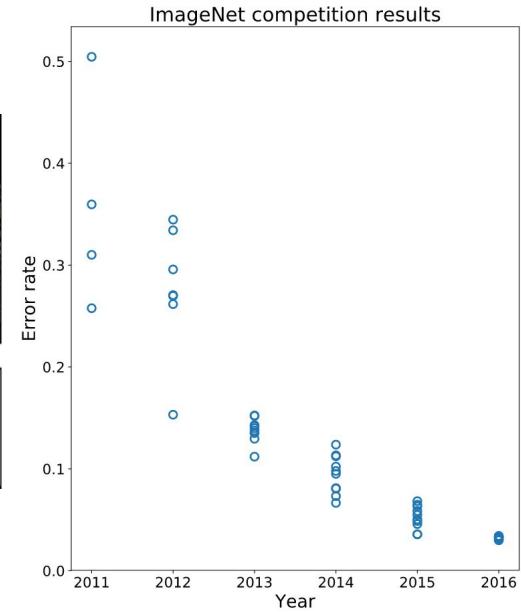
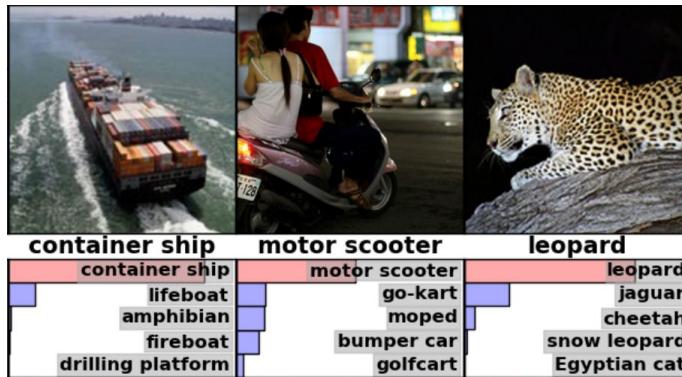
Computer vision is one of the first places where deep learning started to outshine other methods.



Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

2. Image classification



Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

3. Text translation



Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

4. Image generation

Progression in the capabilities of GANs



Brundage et al. "The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation" (2018)



Brock et al. "Large Scale GAN Training for High Fidelity Natural Image Synthesis" (2018)
Introduction to Deep Learning - ICML Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

4. Image generation

Image-to-image translation



Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks" (2016)

Text-to-image translation

this small bird has a pink breast and crown, and black primaries and secondaries.



Reed et al. "Generative Adversarial Text to Image Synthesis" (2016)



Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. **Text generation**
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

5. Text generation

GPT-3-aided poetry!

Human prompt (**bolded**):

Here's a poem by Dr. Seuss about Elon Musk launching rockets with SpaceX, building electric cars at Tesla, selling flame throwers, dating Grimes, fighting the SEC, crushing short sellers, building brain chips to fight AI, and colonizing mars, and tweeting at all hours of day and night. Musk wants to make humans interplanetary to prevent extinction. The poem rhymes every other line with an ABAB structure. The rhyme structure is typical of Dr. Seuss nursery rhymes.

GPT-3 output (with “several hours” of trial and error):

**Elon Musk
By Dr. Seuss**

Once there was a man who really was a Musk. He liked to build robots and rocket ships and such.

He said, “I’m building a car that’s electric and cool. I’ll bet it outsells those Gasoline-burning clunkers soon!”

They sell cool cars that are electric and clean. And then he wanted to go even farther, you see.

The first Tesla, Roadster, Was just the start of a bigger dream. He said, “I’m building a plane that’s electric and green.

That’s not all! That man is also a Musk. He likes electric cars, flamethrowers, tunnels, and such.

But then he thought, “If we make a very small, small chip, We’ll implant it in our heads, And then connect our minds to the Internet!”

...

Source: <https://arr.am/2020/07/14/elon-musk-by-dr-seuss-gpt-3/> (2020)

Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
- 5. Text generation**
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

5. Text generation

GPT-3-generated text:

Human prompt is bolded.

Movie plot summaries (from Wikipedia):

The Matrix (1999): "A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers." (For the purposes of this post, we can set aside some of the specific details and just say that, basically, Neo learns that our world isn't real—that what we perceive isn't the real thing, but a simulation or “matrix” created to fool us.)

Inception (2010): "While competing with a rival thief to recover a potent artefact, Dom Cobb (Leonardo DiCaprio) falls into a dreamworld where he must ‘extract’ an idea from an influential businessman while being chased by the FBI."

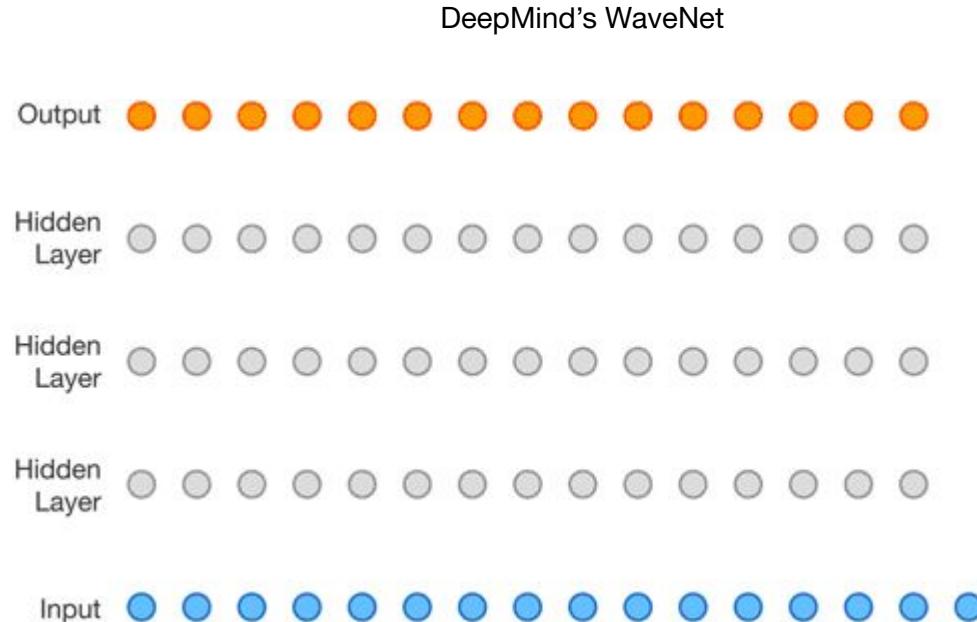
Source: <https://www.gwern.net/GPT-3> (2020)

Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
- 6. Audio generation**
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

6. Audio generation



Source: <https://deepmind.com/blog/article/wavenet-generative-model-raw-audio> (2016)

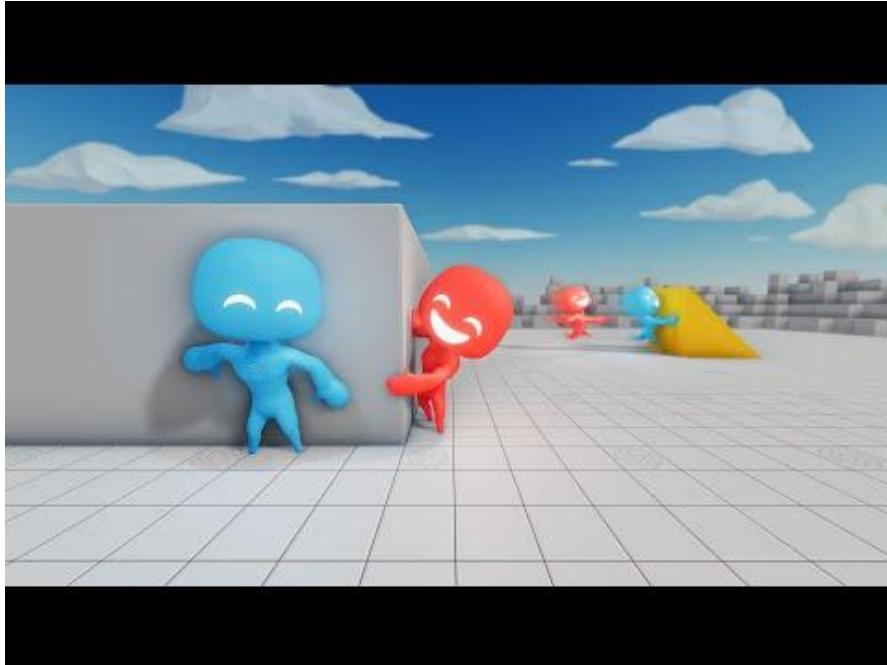
Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

7. Reinforcement learning

OpenAI's "Hide and Seek" Game



Source: <https://openai.com/blog/emergent-tool-use/> (2019)

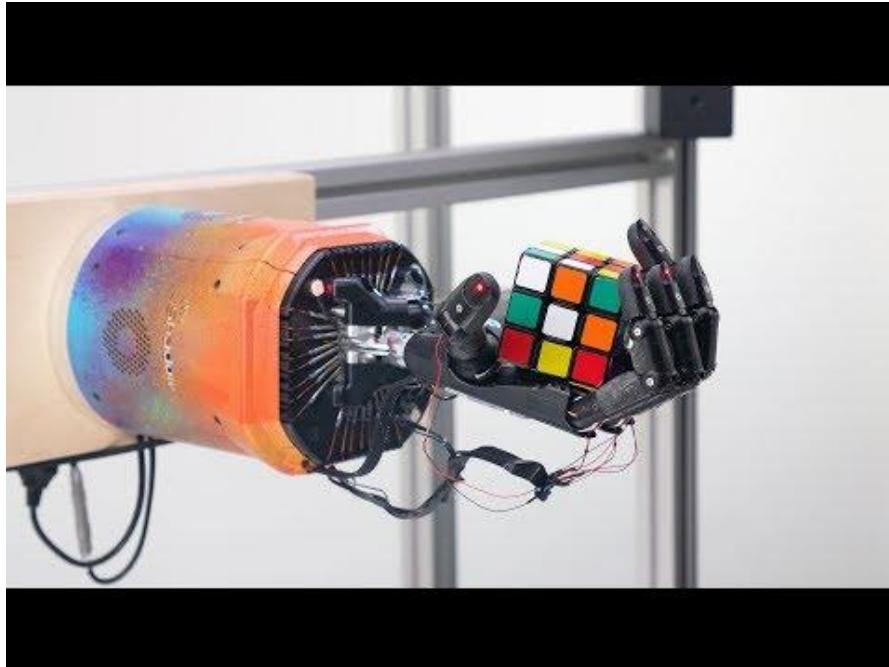
Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

7. Reinforcement learning

Solving a Rubik's cube with a robot hand



Source: <https://openai.com/blog/solving-rubiks-cube/> (2019)

Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

7. Reinforcement learning

OpenAI vs. best human players in DOTA 2



Source: <https://openai.com/projects/five/> (2019)

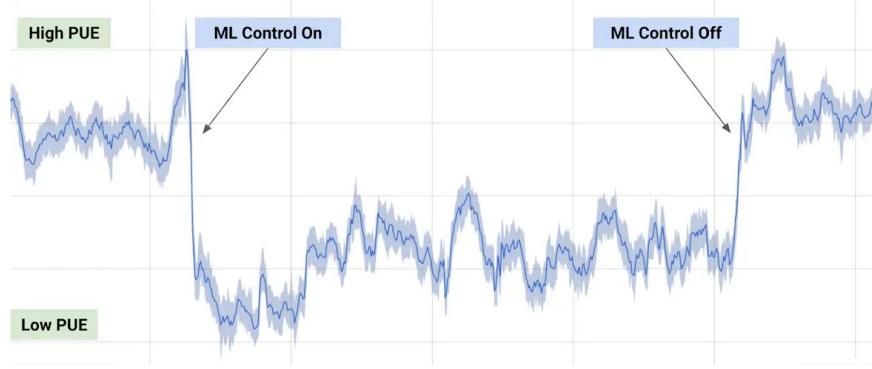
Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

8. Energy

“DeepMind AI Reduces Google Data Centre Cooling Bill by 40%”



MENU ▾

nature

Article | Published: 19 February 2020

Closed-loop optimization of fast-charging protocols for batteries with machine learning

Peter M. Attia, Aditya Grover, Norman Jin, Kristen A. Severson, Todor M. Markov, Yang-Hung Liao, Michael H. Chen, Bryan Cheong, Nicholas Perkins, Zi Yang, Patrick K. Herring, Muratahan Aykol, Stephen J. Harris, Richard D. Braatz✉, Stefano Ermon✉ & William C. Chueh✉

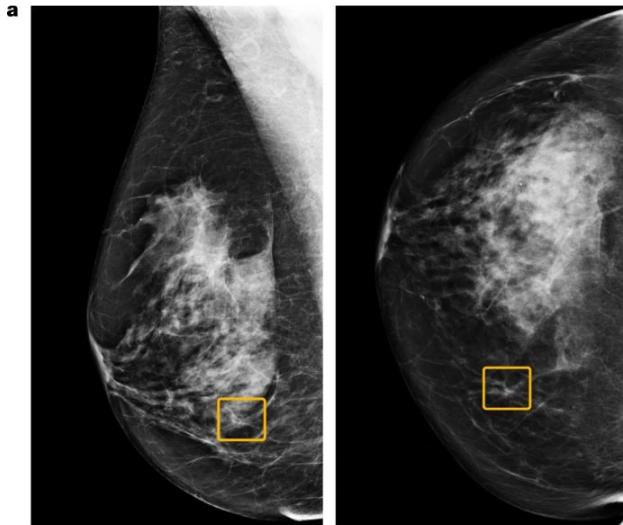
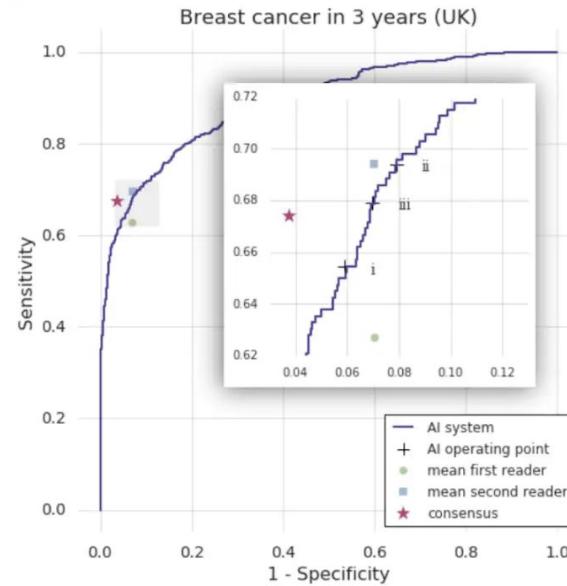
Source: (top) <https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40> (2016)
Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

9. Healthcare

DeepMind's AI for breast cancer screening



Source: McKinney et al. "International evaluation of an AI system for breast cancer screening" (2020)

Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

10. Sustainable development

Estimating poverty using transfer learning and night lights



Source: Jean et al. "Combining satellite imagery and machine learning to predict poverty" (2016)

Introduction to Deep Learning - ICME Summer Workshops 2020

Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
- 11. Investing**
12. Surveillance

11. Investing

Predicting stock price from company fundamentals



Source: Alberg and Lipton. "Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals" (2017)
Introduction to Deep Learning - ICME Summer Workshops 2020

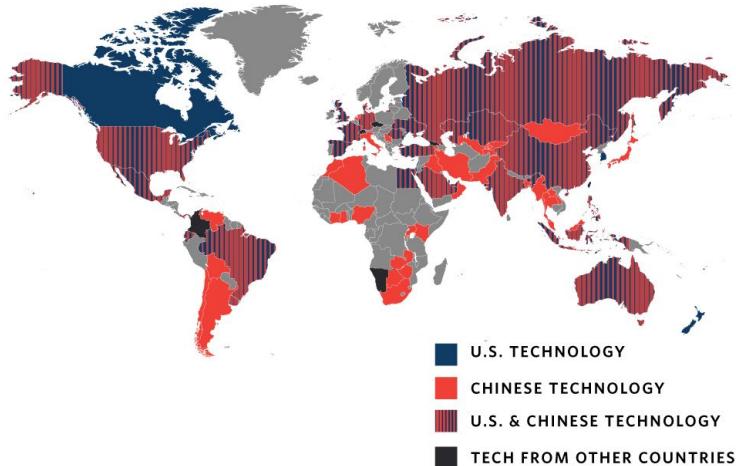
Deep learning examples

1. Optical character recognition (OCR)
2. Image classification
3. Text translation
4. Image generation
5. Text generation
6. Audio generation
7. Reinforcement learning
8. Energy
9. Healthcare
10. Sustainable development
11. Investing
12. Surveillance

12. Surveillance



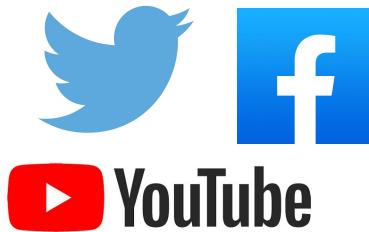
MAP 1
AI Surveillance Technology Origin



Sources: <https://www.nytimes.com/2019/05/14/us/facial-recognition-ban-san-francisco.html> (2019)
<https://carnegieendowment.org/2019/09/17/global-expansion-of-ai-surveillance-pub-79847> (2019)

Why is deep learning at the forefront now?

Big data



amazon.com

Recommended for You

Amazon.com has new recommendations for you based on [items](#) you purchased or told us you own.

Books

- Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop
- Google Apps Administrator Guide: A Private-Label Web Workspace
- Googlepedia: The Ultimate Google Resource (3rd Edition)



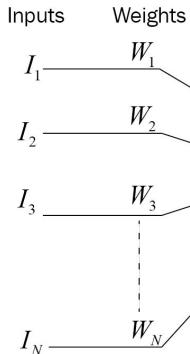
Compute



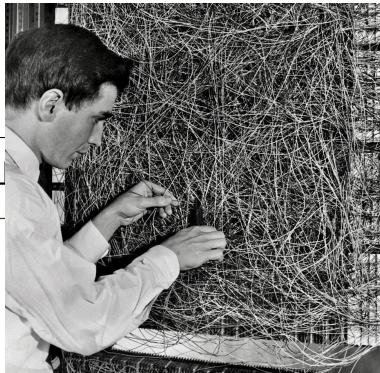
Why is deep learning at the forefront now?

Algorithms

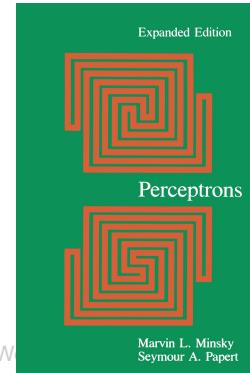
1943: McCulloch and Pitts develop a mathematical model of the neuron



1957: Rosenblatt invents the perceptron algorithm and machine



1969: Minsky and Papert improve the perceptron and prove it cannot model nonlinearities like XOR



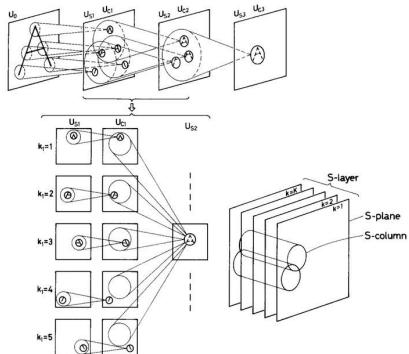
1974–1980: The First AI Winter



Why is deep learning at the forefront now?

Algorithms

1980: Fukushima creates the “neocognitron”, introducing convolutional and downsampling layers

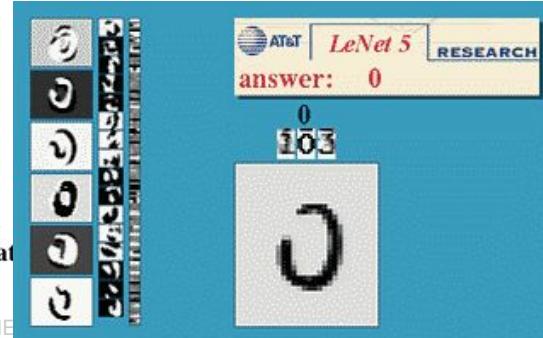


1986: Rumelhart, Hinton, and Williams apply backpropagation to train neural networks



D.E. Rumelhart, G.E. Hinton, R.J. Williams
Learning representation by back-propagated errors. *Nature*, 323 (1986), pp. 533–536

1989: LeCun uses backprop to train convolutional neural networks to read digits



1987–1993:
The Second AI Winter



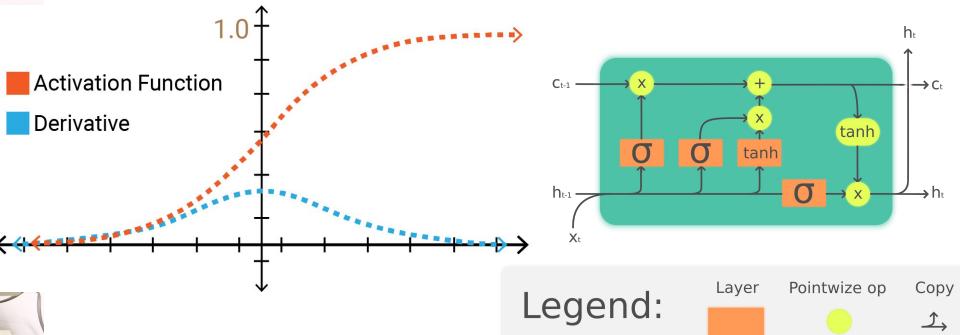
Why is deep learning at the forefront now?

Algorithms

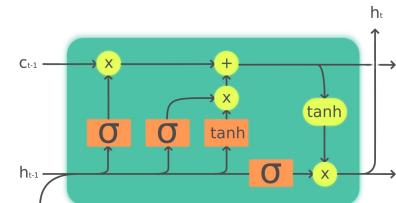
1990s: Computers become faster, GPUs developed



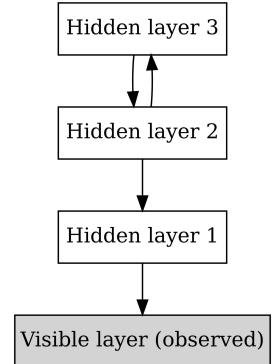
1991: The vanishing gradient problem is identified



1997: Hochreiter and Schmidhuber develop long short-term memory



2006: Hinton and others show that deeper networks can be trained one layer at a time



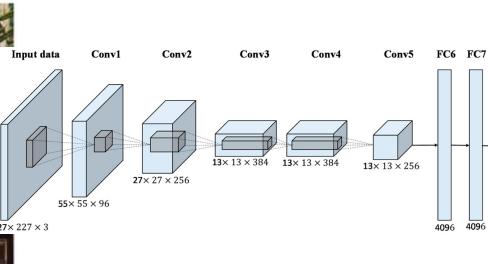
Why is deep learning at the forefront now?

Algorithms

2009: Fei-fei Li releases ImageNet, a classification dataset of 14 million labeled images



2012: AlexNet, a CNN, wins the Large Scale Visual Recognition Challenge — by a lot



2014: Goodfellow designs generative adversarial networks for data generation



2015-2016: Deep learning frameworks like TensorFlow, Keras, and PyTorch are released



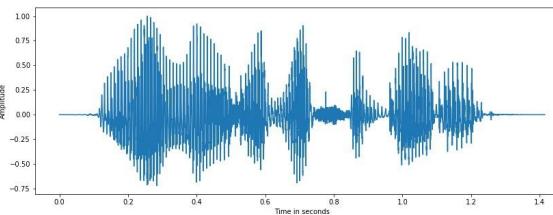
What's so special about deep learning?

Deep learning has made significant improvements in performance on tasks involving unstructured data

Images



Speech

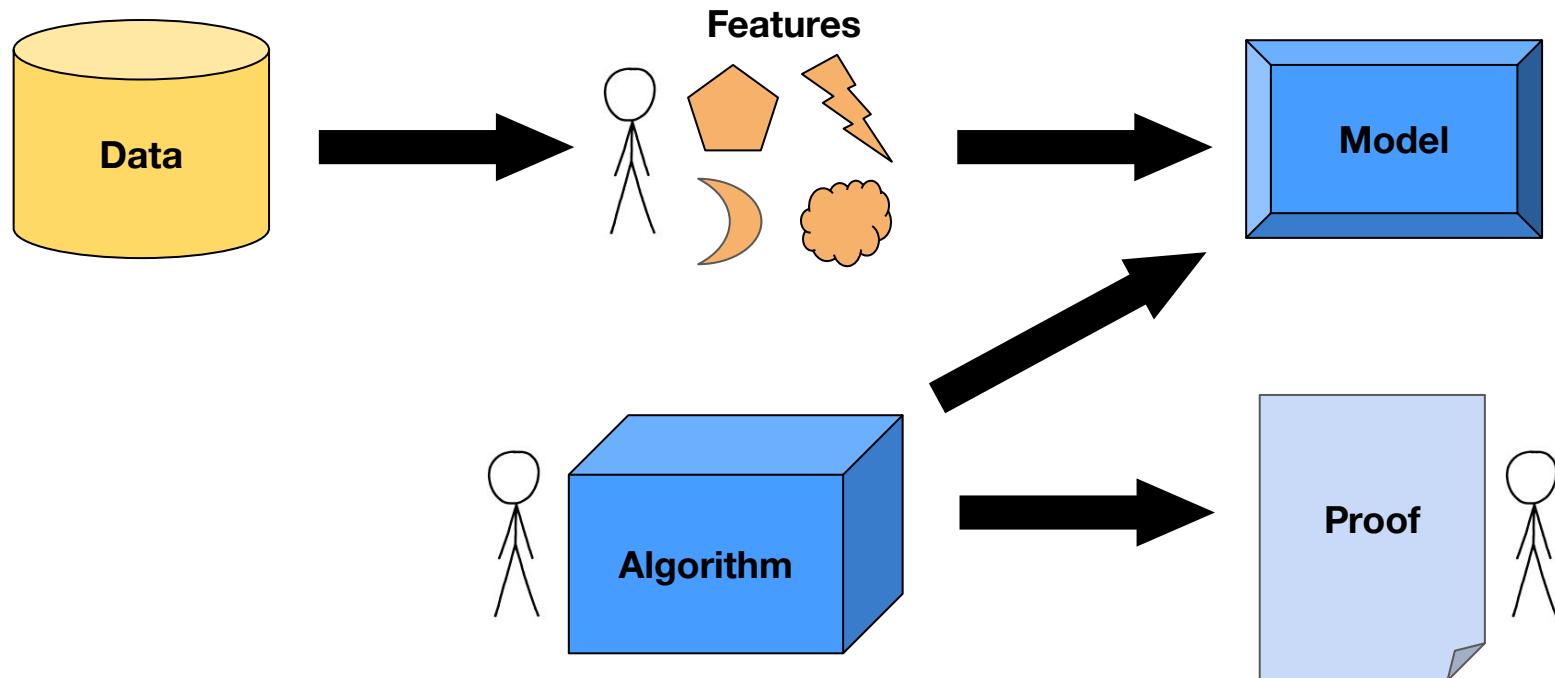


Text

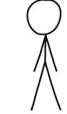
It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife. However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered ...

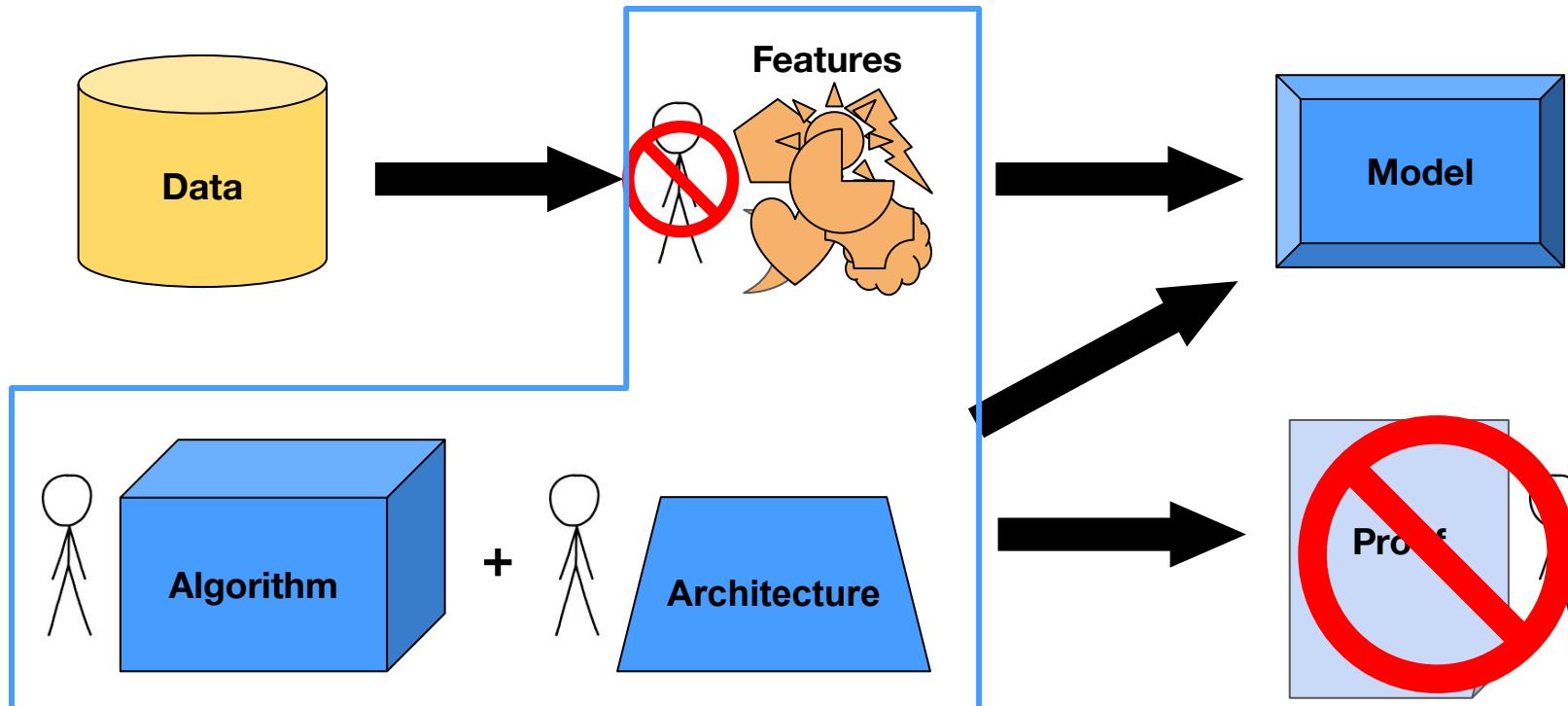
Traditional machine learning

 = Human involved in designing

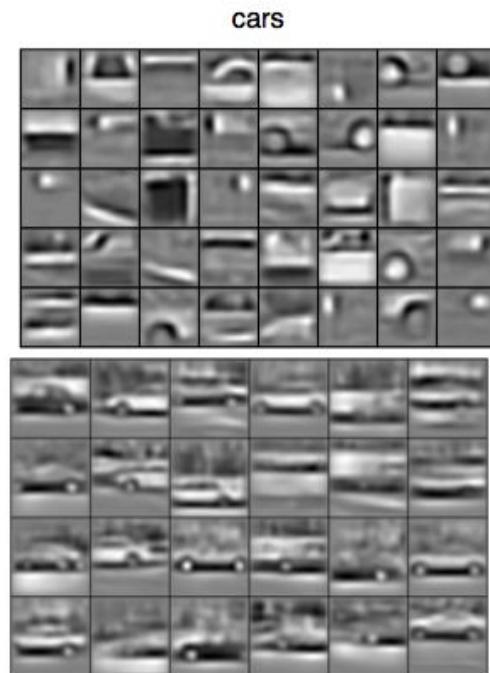
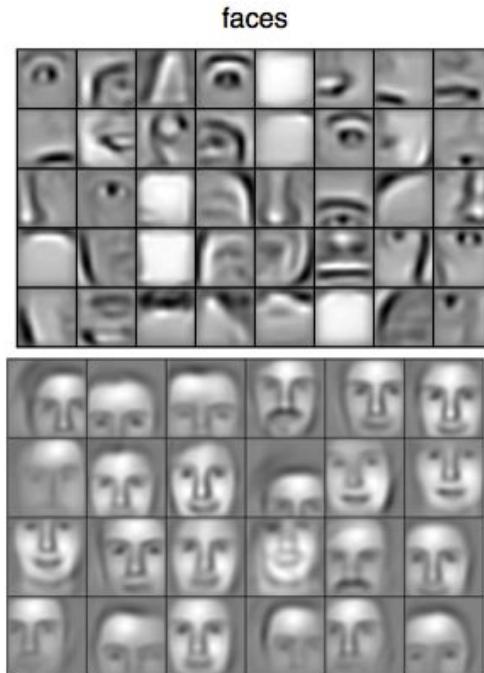


Deep learning

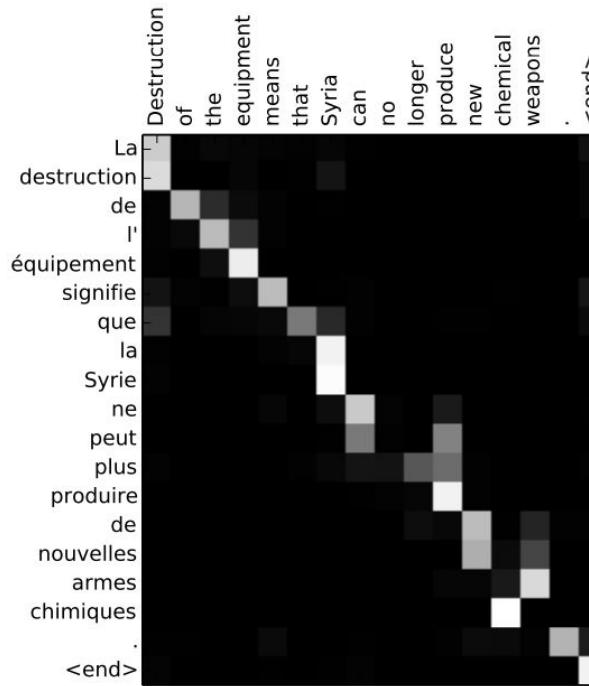
 = Human involved in designing



Deep learning-generated features



Deep learning-generated features



Math Review

Vectors, matrices, and tensors

What is a vector?

$$\begin{bmatrix} 2 \\ 7 \\ 3 \end{bmatrix}$$

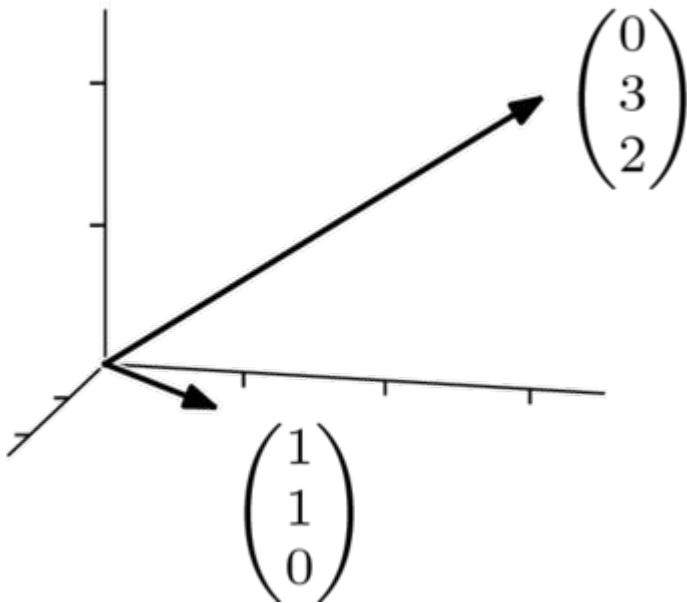


An array (or list) of numbers

Dimension = 3

$$\vec{v} \in \mathbb{R}^3$$

What is a vector?



In addition to linear algebra, you may have learned about vectors in physics courses as entities with direction and magnitude

Here are two more vectors of dimension 3, visualized in Euclidean space

Vector operations

1. Scalar multiplication

$$4 * \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix} = \begin{pmatrix} 8 \\ 28 \\ 12 \end{pmatrix}$$

Vector operations

2. Addition

$$\begin{bmatrix} 2 \\ 7 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \\ 6 \end{bmatrix} = \begin{bmatrix} 3 \\ 10 \\ 9 \end{bmatrix}$$

Vector operations

3. Transpose

$$\begin{bmatrix} 2 \\ 7 \\ 3 \end{bmatrix}^T = \begin{bmatrix} 2 & 7 & 3 \end{bmatrix}$$

A diagram illustrating the transpose operation. On the left, a vertical column vector is enclosed in a blue oval. It has three entries: 2, 7, and 3, arranged vertically. A blue arrow points from this oval to the text "Column vector". Above the equals sign is a superscript T, indicating the transpose. To the right of the equals sign is a horizontal row vector enclosed in a red oval. It also has three entries: 2, 7, and 3, arranged horizontally. A red arrow points from this oval to the text "Row vector".

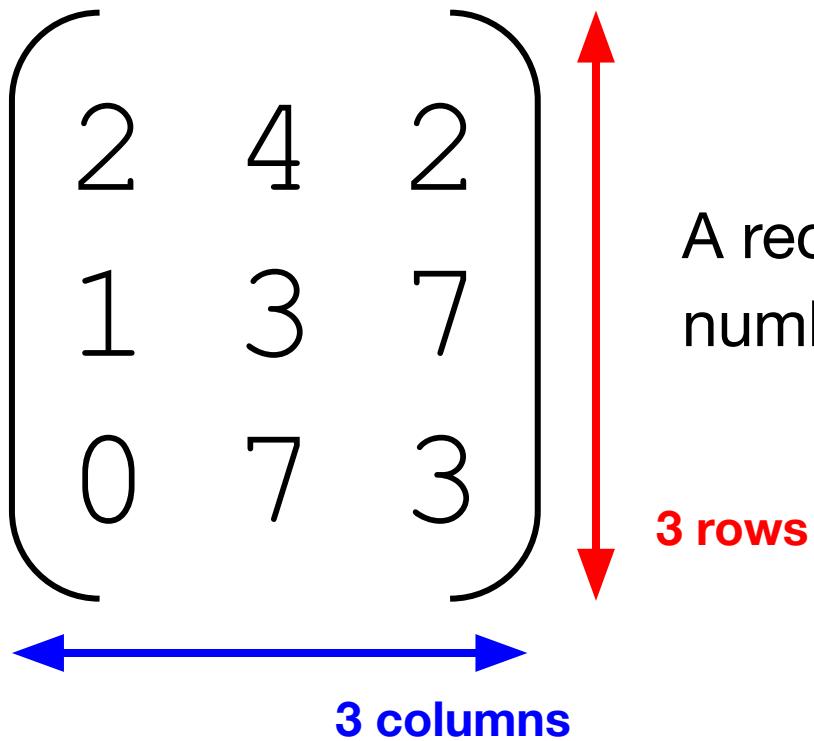
Vector operations

4. Inner product

$$\begin{pmatrix} 2 & 7 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix} = 2 * 1 + 7 * 3 + 3 * 6 = 41$$

$\vec{u}^\top \vec{v} = w$

What is a matrix?



A rectangular array (or table) of numbers

What is a matrix?

$$\begin{matrix} & \begin{matrix} 1 & 2 & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ m \end{matrix} & \left[\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right] \end{matrix} \quad \mathbf{A} \in \mathbb{R}^{m \times n}$$

What is a matrix?

$$\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix} \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix}$$

You can think of a matrix as a row vector of column vectors

What is a matrix?

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix}$$

... or as a column vector of row vectors

Matrix operations

1. Scalar multiplication
2. Addition

$$2 * \begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} = \begin{pmatrix} 4 & 8 & 4 \\ 2 & 6 & 14 \\ 0 & 14 & 6 \end{pmatrix}$$

Matrix operations

3. Transpose

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Matrix operations

3. Transpose

$$\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix} \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix}^T = \begin{pmatrix} 2 & 1 & 0 \\ 4 & 3 & 7 \\ 2 & 7 & 3 \end{pmatrix}$$

Matrix operations

3. Transpose

$$\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix} \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix}^T = \begin{pmatrix} 2 & 1 & 0 \\ 4 & 3 & 7 \\ 2 & 7 & 3 \end{pmatrix}$$

Matrix operations

4. Multiplication with a vector

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 18 \\ 24 \\ 27 \end{pmatrix}$$

Matrix operations

4. Multiplication with a vector

$$\begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix} \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 18 \\ 24 \\ 27 \end{pmatrix}$$

Matrix operations

4. Multiplication with a vector

$$1 \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + 3 \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix} + 2 \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix} = \begin{pmatrix} 18 \\ 24 \\ 27 \end{pmatrix}$$

Matrix operations

4. Multiplication with a vector

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 18 \\ 24 \\ 27 \end{pmatrix}$$

Matrix operations

4. Multiplication with a vector

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \\ 1 \\ 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 18 \\ 24 \\ 27 \end{pmatrix}$$

Matrix operations

4. Multiplication with a vector

$$\begin{pmatrix} \vec{a}_1^\top \\ \vec{a}_2^\top \\ \vec{a}_3^\top \end{pmatrix} \vec{b} = \begin{pmatrix} 18 \\ 24 \\ 27 \end{pmatrix}$$

Matrix operations

4. Multiplication with a vector

$$\begin{pmatrix} \vec{a}_1^\top \vec{b} \\ \vec{a}_2^\top \vec{b} \\ \vec{a}_3^\top \vec{b} \end{pmatrix} = \begin{pmatrix} 18 \\ 24 \\ 27 \end{pmatrix}$$

Matrix operations

4. Multiplication with another matrix

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} \begin{pmatrix} 1 & 9 \\ 3 & 0 \\ 2 & 4 \end{pmatrix} = \begin{pmatrix} 18 & 26 \\ 24 & 37 \\ 27 & 12 \end{pmatrix}$$

Diagram illustrating matrix multiplication:

- The first matrix has 3 columns (indicated by a blue double-headed arrow labeled "3 columns").
- The second matrix has 2 rows (indicated by red double-headed arrows labeled "3 rows").
- The result matrix has 2 columns (indicated by a blue double-headed arrow labeled "2 columns").
- The result matrix has 3 rows (indicated by red double-headed arrows labeled "3 rows").

Red arrows indicate the rows of the second matrix being multiplied with the columns of the first matrix.

Matrix operations

4. Multiplication with another matrix

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 18 & 26 \\ 24 & 37 \\ 27 & 12 \end{pmatrix}$$

Matrix operations

4. Multiplication with another matrix

$$A \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \begin{pmatrix} 9 \\ 0 \\ 4 \end{pmatrix} = \begin{pmatrix} A \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \\ A \begin{pmatrix} 9 \\ 0 \\ 4 \end{pmatrix} \end{pmatrix}$$

Matrix operations

4. Multiplication with another matrix

$$\begin{array}{ccc} & \vec{b}_1 & \vec{b}_2 \\ & \downarrow & \downarrow \\ \vec{a}_1 \rightarrow & \begin{bmatrix} 1 & 7 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} 3 & 3 \\ 5 & 2 \end{bmatrix} & = \begin{bmatrix} \vec{a}_1 \cdot \vec{b}_1 & \vec{a}_1 \cdot \vec{b}_2 \\ \vec{a}_2 \cdot \vec{b}_1 & \vec{a}_2 \cdot \vec{b}_2 \end{bmatrix} \\ \vec{a}_2 \rightarrow & & \\ A & B & C \end{array}$$

Source: https://ml-cheatsheet.readthedocs.io/en/latest/linear_algebra.html#matrix-multiplication

Introduction to Deep Learning - ICME Summer Workshops 2020

Poll: Matrix multiplication

Go to PollEv.com/dlworkshop2020

What are the dimensions of the matrix resulting from:

$$\left(\begin{array}{ccccccc} 2 & 4 & 2 & 8 & 4 & 1 & 5 & 0 \\ 1 & 3 & 7 & 9 & 3 & 2 & 6 & 8 \\ 0 & 7 & 3 & 1 & 0 & 4 & 9 & 3 \end{array} \right) \quad \left(\begin{array}{c} 3 \\ 2 \\ 0 \\ 5 \\ 3 \\ 6 \\ 2 \\ 4 \end{array} \right)$$

- A. 8×8
- B. 8×2
- C. 3×2
- D. Not valid

What is a tensor?

A tensor is an N-dimensional array of data



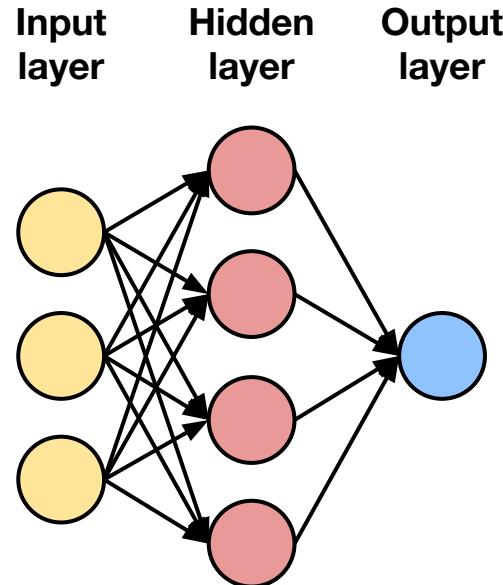
Neural Networks

Neural networks

We'll start with a *fully connected* neural network.

1. Single layer
2. Single neuron
3. Multiple layers
4. Input and output

Later, we'll cover other types of layers (convolution, max-pool, ...)



What does a single layer in a neural network do?

First, it involves some matrix operations...

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} + \begin{pmatrix} -3 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 15 \\ 24 \\ 29 \end{pmatrix}$$

What does a single layer in a neural network do?

Then, the intermediate values are passed through a nonlinear function...

$$g \left(\begin{pmatrix} 15 \\ 24 \\ 29 \end{pmatrix} \right) = \begin{pmatrix} g(15) \\ g(24) \\ g(29) \end{pmatrix}$$

Activation function

Activation functions

There exist a number of activation functions, some of which were designed with neurons in mind

What's the purpose?

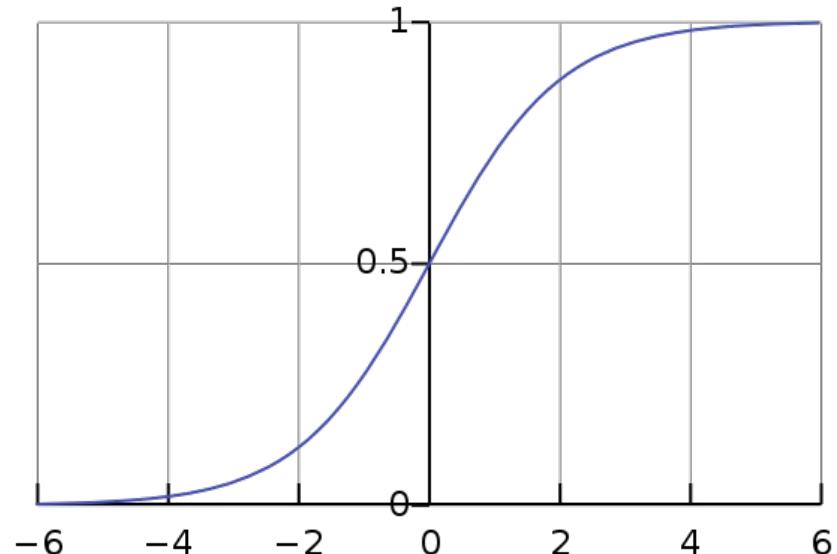
- Most phenomena are not linear, so we introduce some nonlinearities to the network

"A visual proof that neural nets can compute any function":
<http://neuralnetworksanddeeplearning.com/chap4.html>

| Activation function | Equation | Example | 1D Graph |
|--|---|--|----------|
| Unit step (Heaviside) | $\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$ | Perceptron variant | |
| Sign (Signum) | $\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$ | Perceptron variant | |
| Linear | $\phi(z) = z$ | Adaline, linear regression | |
| Piece-wise linear | $\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$ | Support vector machine | |
| Logistic (sigmoid) | $\phi(z) = \frac{1}{1 + e^{-z}}$ | Logistic regression, Multi-layer NN | |
| Hyperbolic tangent | $\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ | Multi-layer Neural Networks | |
| Rectifier, ReLU (Rectified Linear Unit) | $\phi(z) = \max(0, z)$ | Multi-layer Neural Networks | |
| Rectifier, softplus | $\phi(z) = \ln(1 + e^z)$ | Multi-layer Neural Networks | |

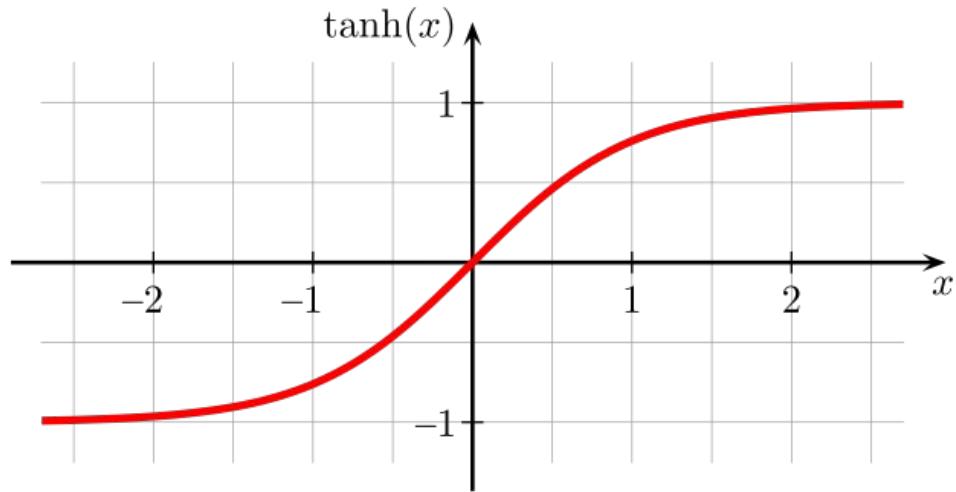
Sigmoid (logistic) activation function

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$



Hyperbolic tangent activation function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



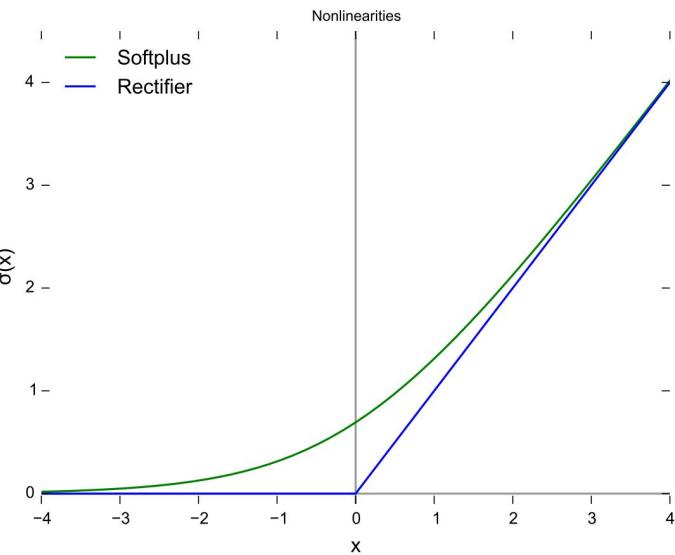
ReLU activation function

Stands for **R**ectified **L**inear **U**nit

$$\text{ReLU}(x) = \max(0, x)$$

A smooth version of ReLU is softplus

$$\text{softplus}(x) = \ln(1 + e^x)$$



Poll: Activation functions

Go to PollEv.com/dlworkshop2020

As its input goes to infinity, the output of the logistic function goes to...

$$\lim_{x \rightarrow \infty} \frac{1}{1 + e^{-x}} =$$

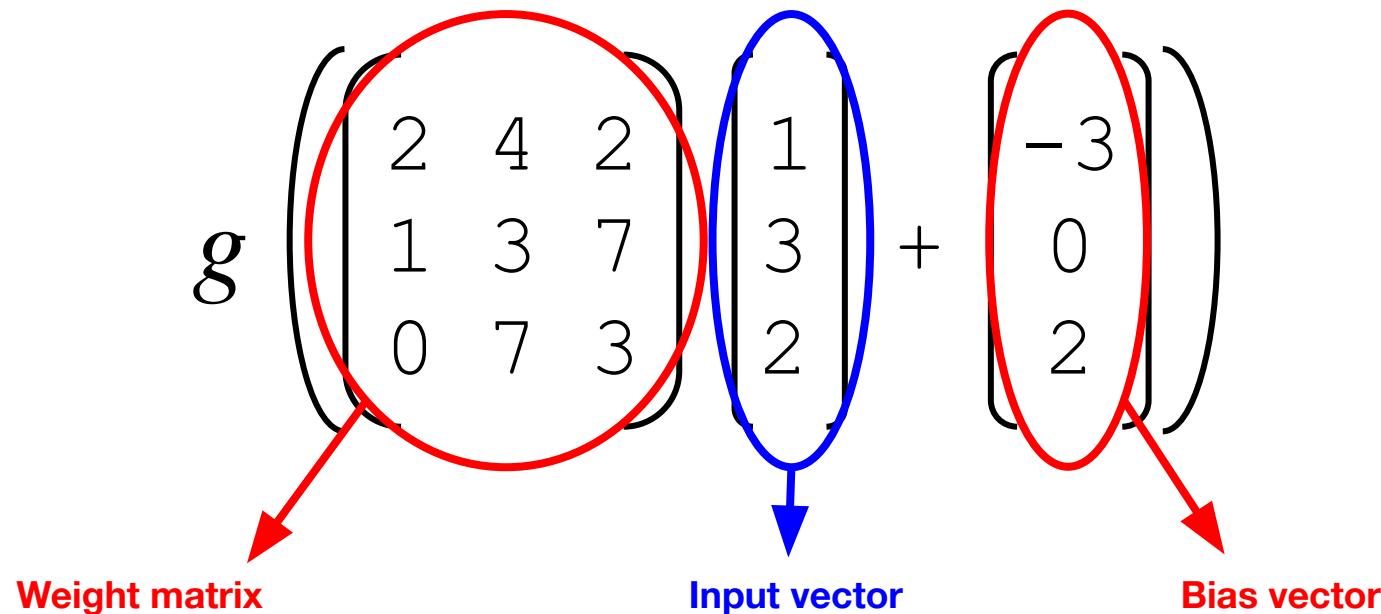
- A. 0
- B. 0.5
- C. 1
- D. Infinity

Activation functions FAQ

1. Why these activation functions? Why not x^2 or other nonlinear functions?
2. How does one choose among the various options for activation functions?

What are the matrix and vectors?

Putting the linear operations and nonlinearity together, we get:



What are the matrix and vectors?

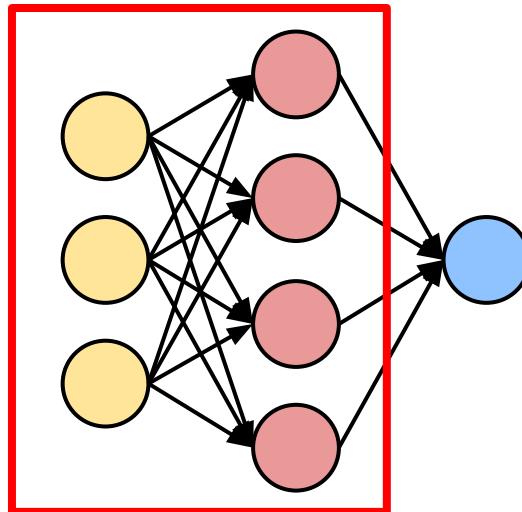
Putting the linear operations and nonlinearity together, we get:

$$g \left(\begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} + \begin{pmatrix} -3 \\ 0 \\ 2 \end{pmatrix} \right)$$

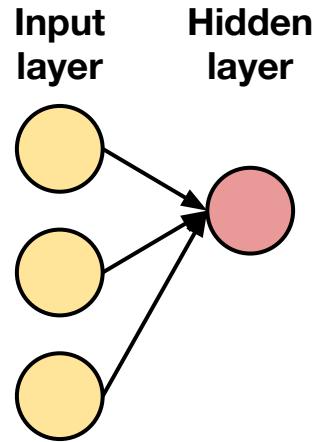
Trainable parameters

A more compact notation

$$g(\mathbf{W}\vec{x} + \vec{b})$$



What does a single neuron do?



Each circle is one element of an input or hidden layer

Each edge is a weight

$$g(\vec{w}_1^\top \vec{x} + b_1)$$

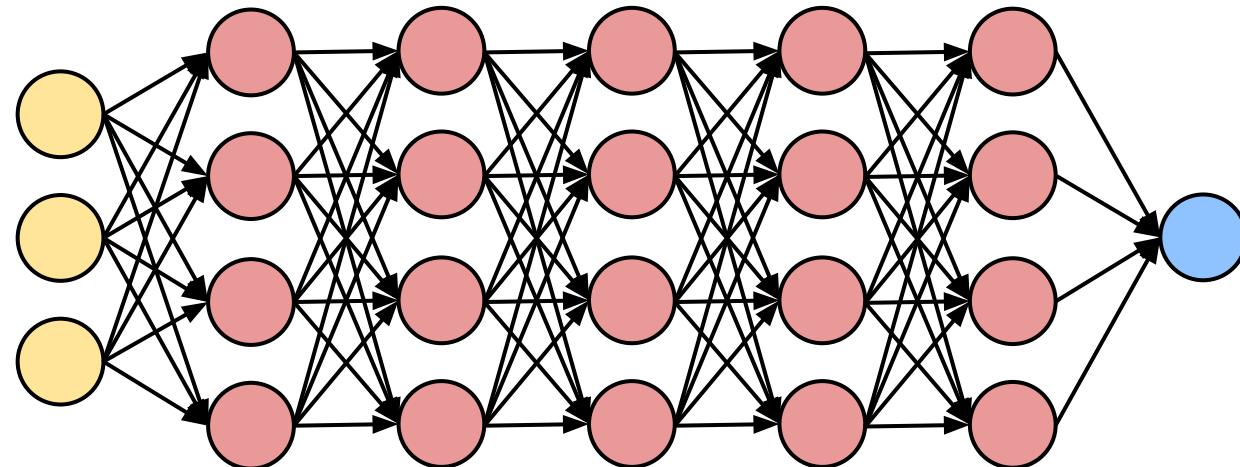
Weights for each neuron

$$\mathbf{W} = \begin{pmatrix} 2 & 4 & 2 \\ 1 & 3 & 7 \\ 0 & 7 & 3 \end{pmatrix} = \begin{pmatrix} w_1^\top \\ w_2^\top \\ w_3^\top \end{pmatrix}$$

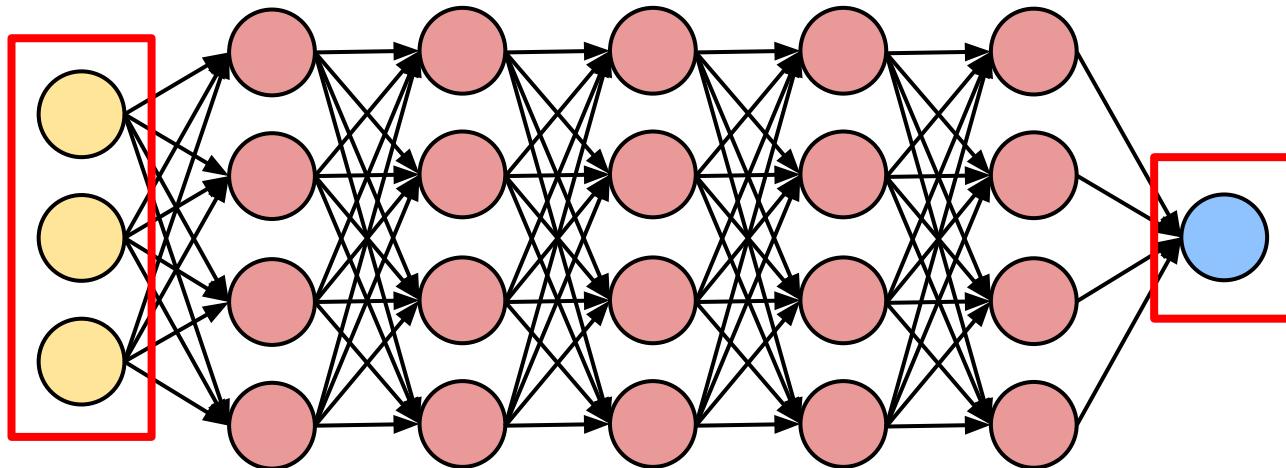
A fully connected layer with weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$
is a function from \mathbb{R}^n to \mathbb{R}^m

Stacking multiple layers

It becomes a *deep* neural network when you use many layers



What are the inputs and outputs?



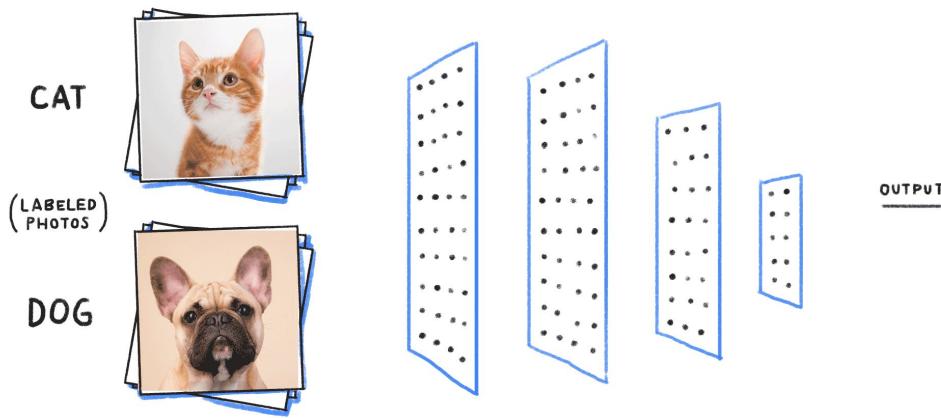
Defining the problem

- You want to predict some output \mathbf{y} given input \mathbf{x}
- If \mathbf{y} is a discrete output (e.g. image classes), then it's a *classification problem*
- If \mathbf{y} is a continuous variable (e.g. price), then it's a *regression problem*

$$y = \text{DNN}(\vec{x})$$

Defining the problem

- If y is a discrete output, you can use **one-hot encoding**
- If y is a continuous variable, it can be output by the network directly



Classes = {cat, dog}

$$y = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

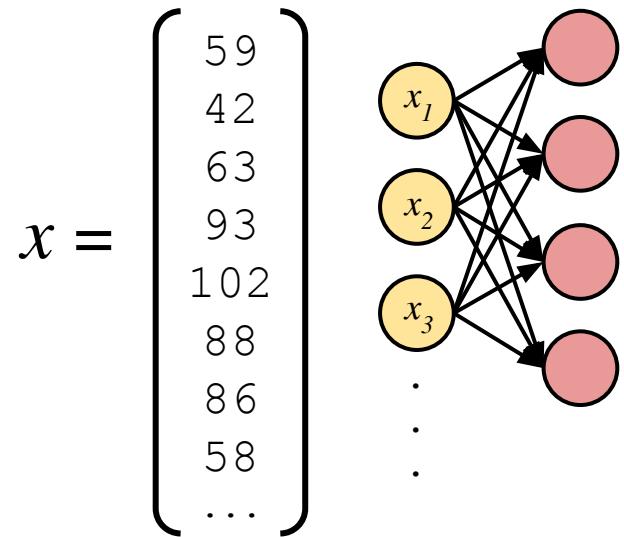
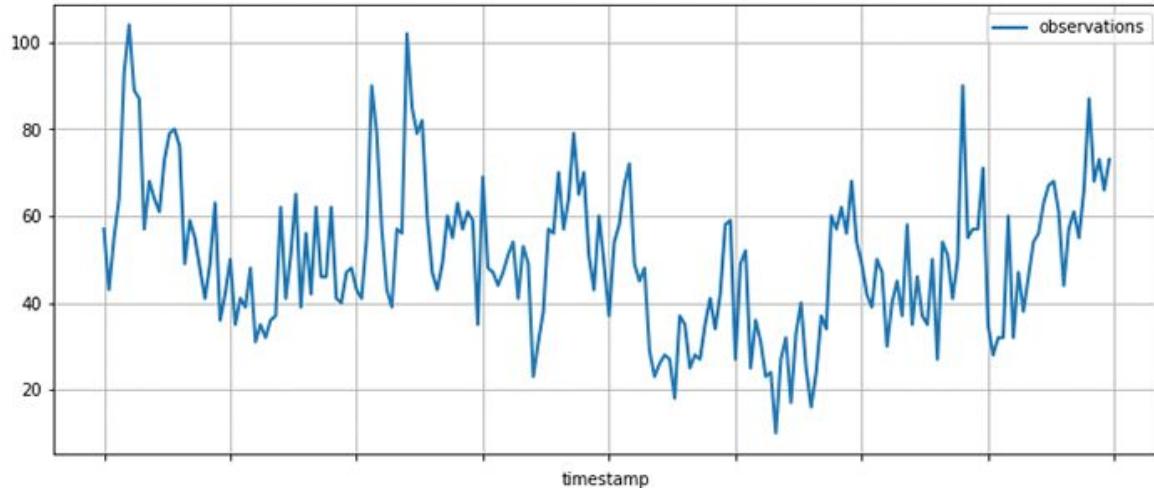
“cat”

$$y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

“dog”

Defining the problem

- If x is a discrete output, you can use **one-hot encoding**
- If x is a continuous variable, it can be input to the network directly

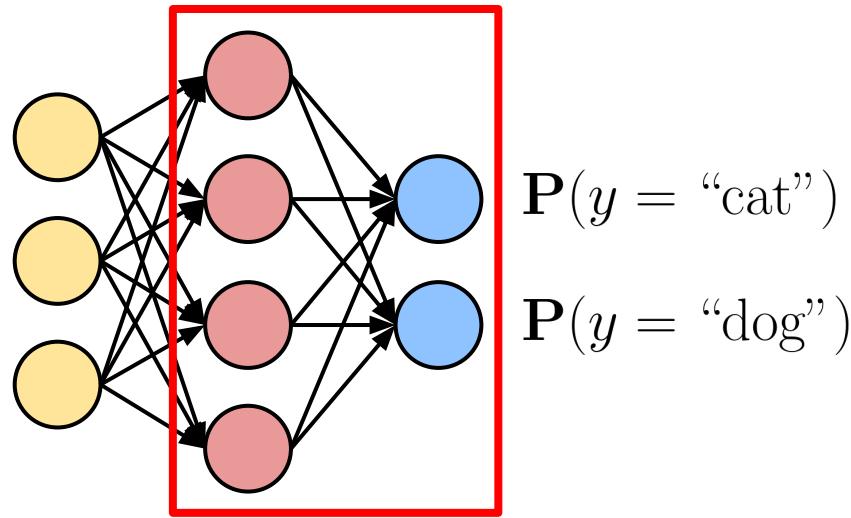


Last layer activation function

For classification, each neuron in the output represents one output class

You want the output to represent the probability that the input belongs to each class

What constraints must the output satisfy?



Last layer activation function

What constraints must the output satisfy?

- Probabilities are positive
- Probabilities sum to 1

$$\begin{aligned} \mathbf{P}(y = \text{"cat"}) &\geq 0 \\ \mathbf{P}(y = \text{"dog"}) &\geq 0 \end{aligned}$$

$$\mathbf{P}(y = \text{"cat"}) + \mathbf{P}(y = \text{"dog"}) = 1$$

Softmax activation function

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}}$$

Ensures positivity
Normalizes sum to 1

$$\mathbf{z} = \begin{pmatrix} 1.1 \\ 0.4 \\ -8.2 \\ -0.9 \\ 3.7 \end{pmatrix}$$

$$e^{\mathbf{z}} = \begin{pmatrix} e^{1.1} \\ e^{0.4} \\ e^{-8.2} \\ e^{-0.9} \\ e^{3.7} \end{pmatrix}$$

$$\frac{e^{\mathbf{z}}}{\sum_{j=1}^n e^{z_j}} = \begin{pmatrix} e^{1.1}/\sum_j e^{\mathbf{z}_j} \\ e^{0.4}/\sum_j e^{\mathbf{z}_j} \\ e^{-8.2}/\sum_j e^{\mathbf{z}_j} \\ e^{-0.9}/\sum_j e^{\mathbf{z}_j} \\ e^{3.7}/\sum_j e^{\mathbf{z}_j} \end{pmatrix} = \begin{pmatrix} 0.07 \\ 0.03 \\ 0.00 \\ 0.01 \\ 0.89 \end{pmatrix}$$

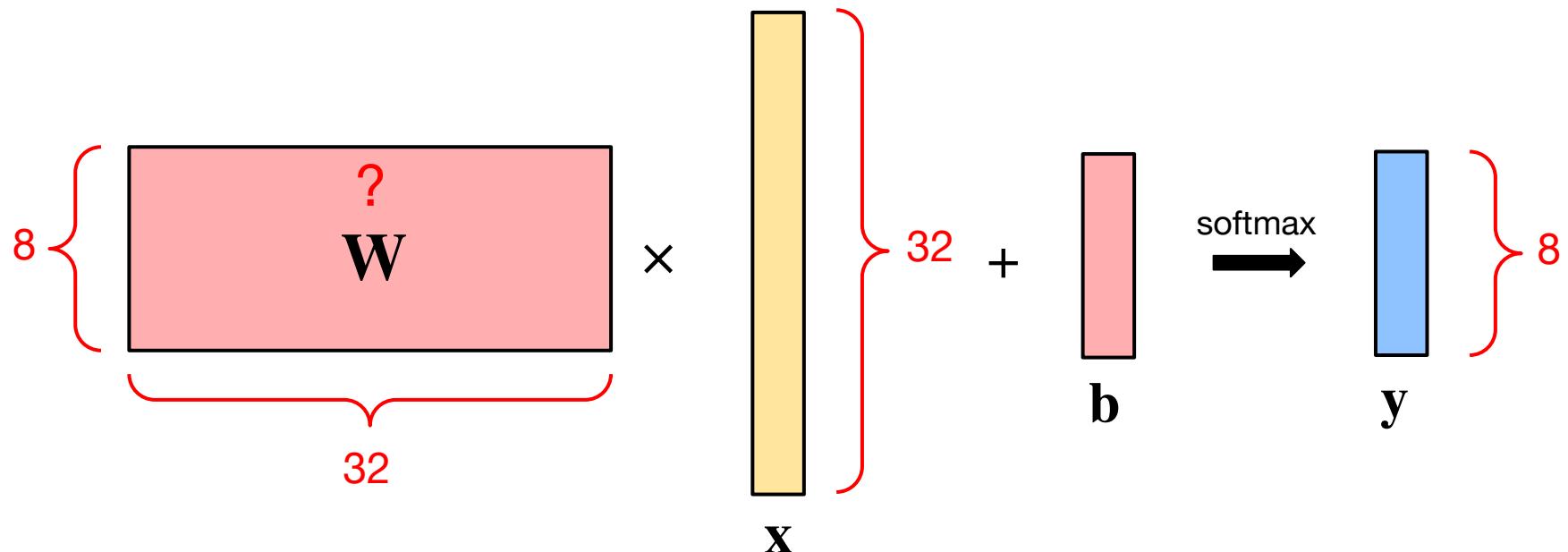
Poll: Weight matrix

Go to PollEv.com/dlworkshop2020

Suppose we have a neural network with 1 fully connected hidden layer. The input is a 32-dimensional vector. The output is a probability distribution across 8 classes. The weight matrix of the hidden layer should have dimension...

- A. 32x1
- B. 8x32
- C. 32x8
- D. 32x32

Poll: Weight matrix



Up Next:
Training Neural Networks