# Mathematics, algebra and AI

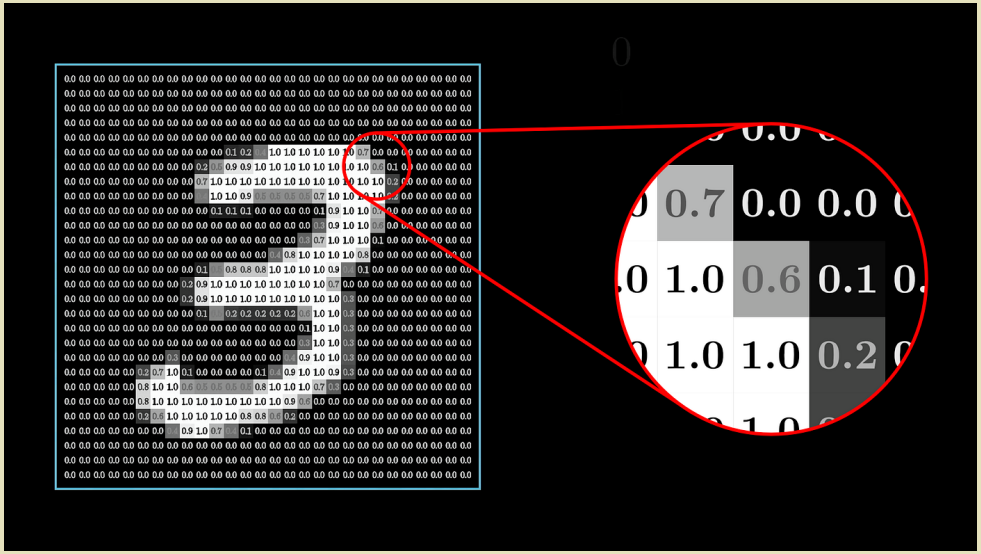
Or: The snake that eats itself

~~Accept~~Change what you cannot ~~change~~accept

# Algebra in AI

- Task Identify handwritten digits
- We can see this as a function in the following way:
  - Convert the pictures into grayscale values, e.g. $28 \times 28$ grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$

## Input example



## Output example

| | |
|---|---|
| 0.1 | 0 |
| 0.0 | 1 |
| 0.0 | 2 |
| 0.6 | 3 |
| 0.0 | 4 |
| 0.0 | 5 |
| 0.1 | 6 |
| 0.0 | 7 |
| 0.0 | 8 |
| 0.2 | 9 |

$\Rightarrow$ with 60% certainty this is a 3

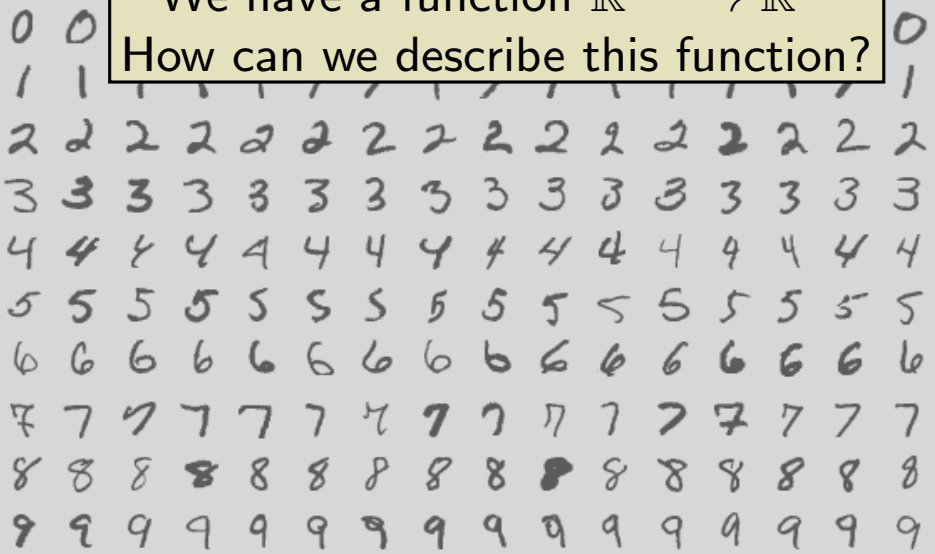
- Task Identify
- We can see thi
  - Convert th
  - Flatten the
  - The outpu
- We thus have a

28 grid of numbers

ith $28^2 = 784$ entries

**Task – rephrased**

We have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
How can we describe this function?

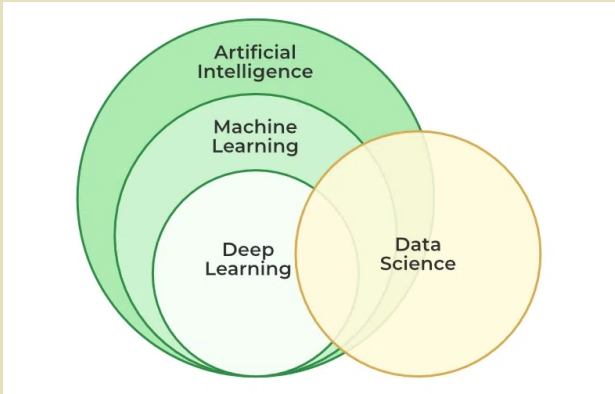


- Task Identify handwritten digits
- We can see this as a function in the following way:
  - Convert the pictures into grayscale values, e.g. $28 \times 28$ grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$

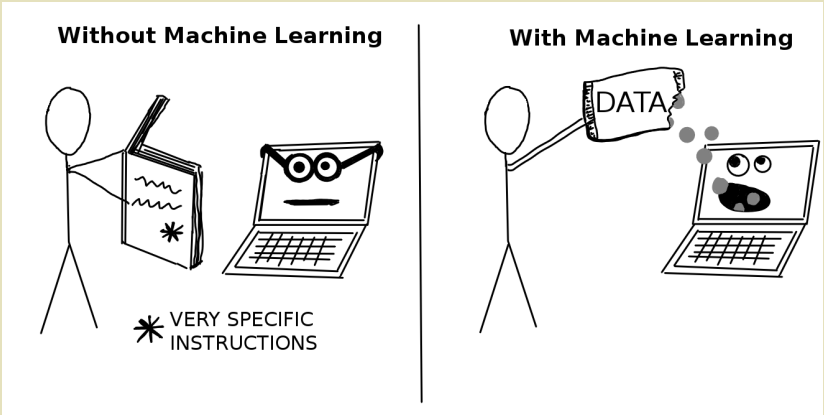
**Task – rephrased**

We have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
How can we describe this function?

**Machine/deep learning** (today's topic)

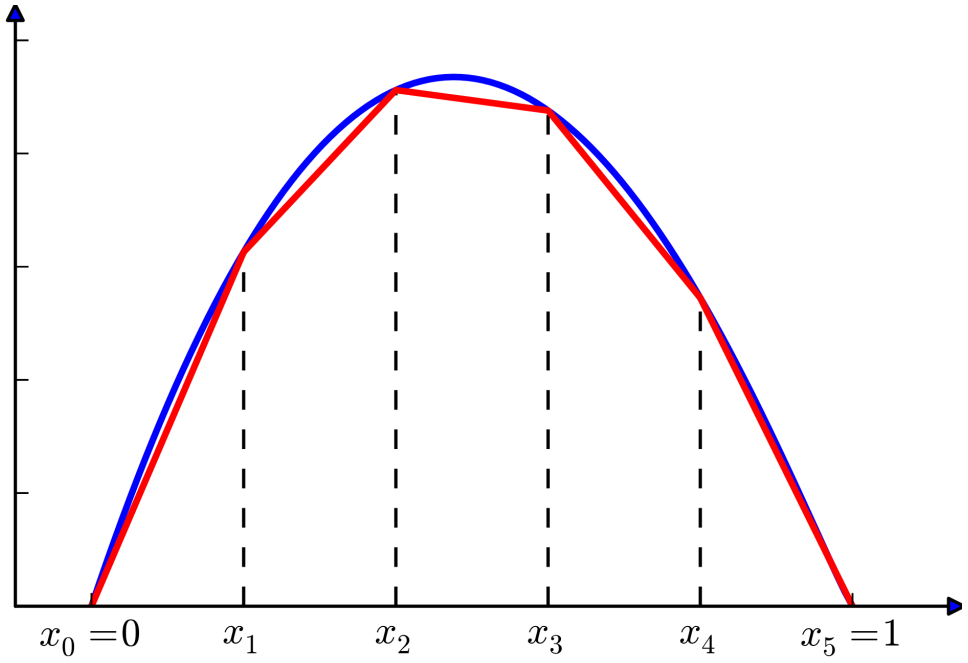



tries to answer questions of this type
letting a computer detect patterns in data

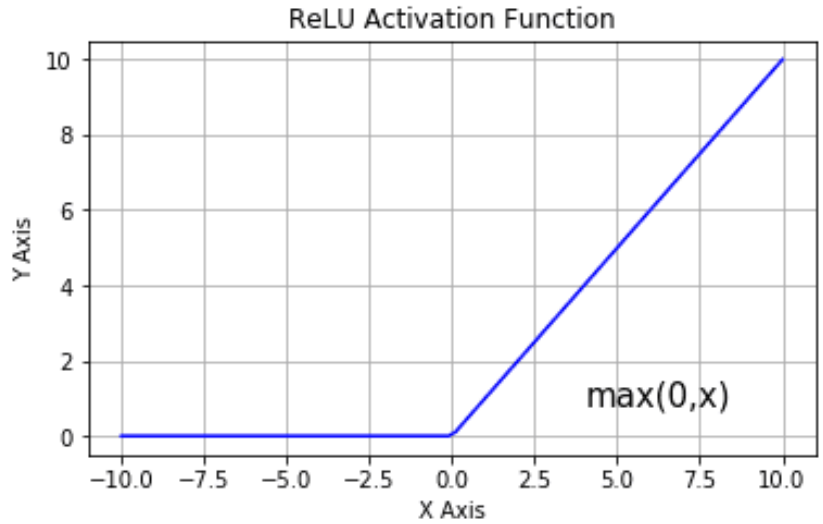



**Crucial** In ML the computer performs tasks without explicit instructions

# Algebra in AI

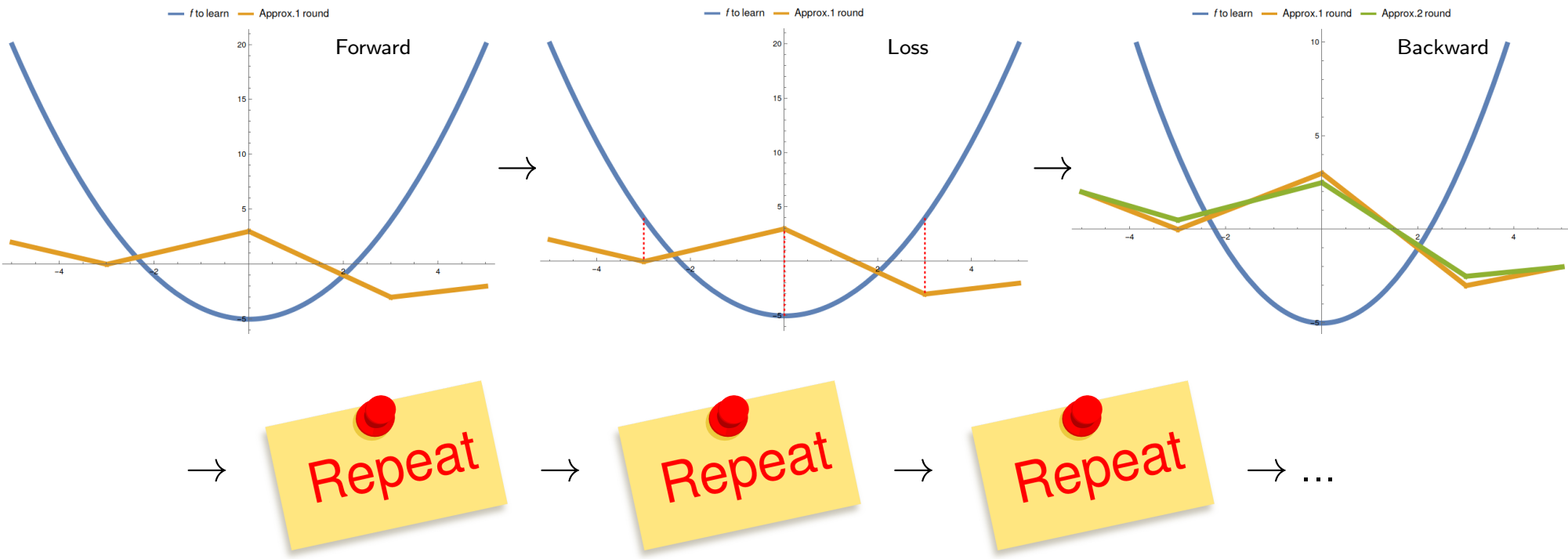



,



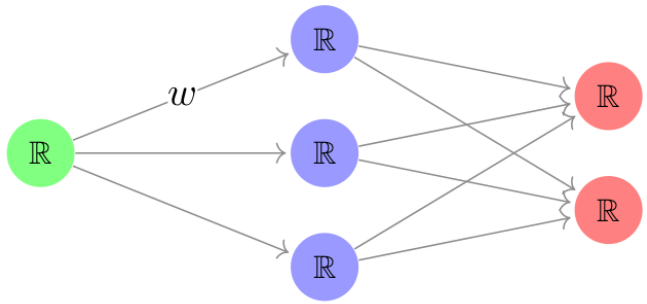

- **Idea** Approximate the unknown function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
- **Neural network** = a piecewise linear approximation (matrices + PL maps)
- The **matrices** = a bunch of numbers (weights) and offsets (biases)
- The **PL maps** = usually ReLU

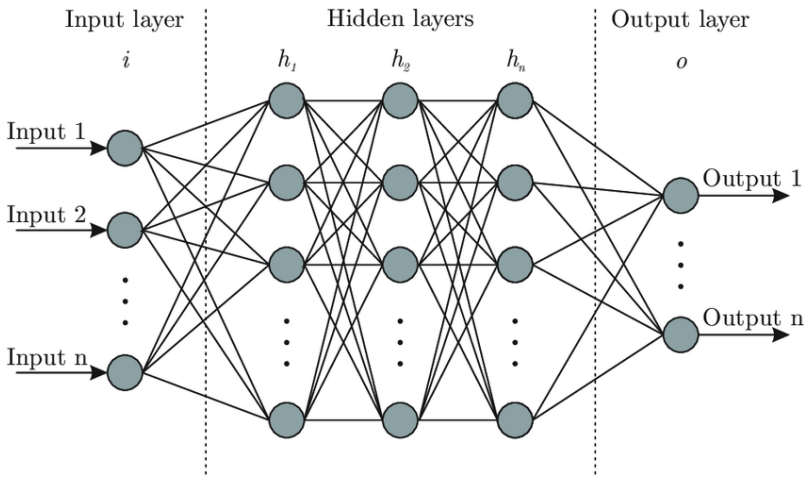
# Algebra in AI




- Machine learning mantra
  - Forward = calculate an approximation (start with random inputs)
  - Loss = compare to real data
  - Backward = adjust the approximation

# Algebra in AI

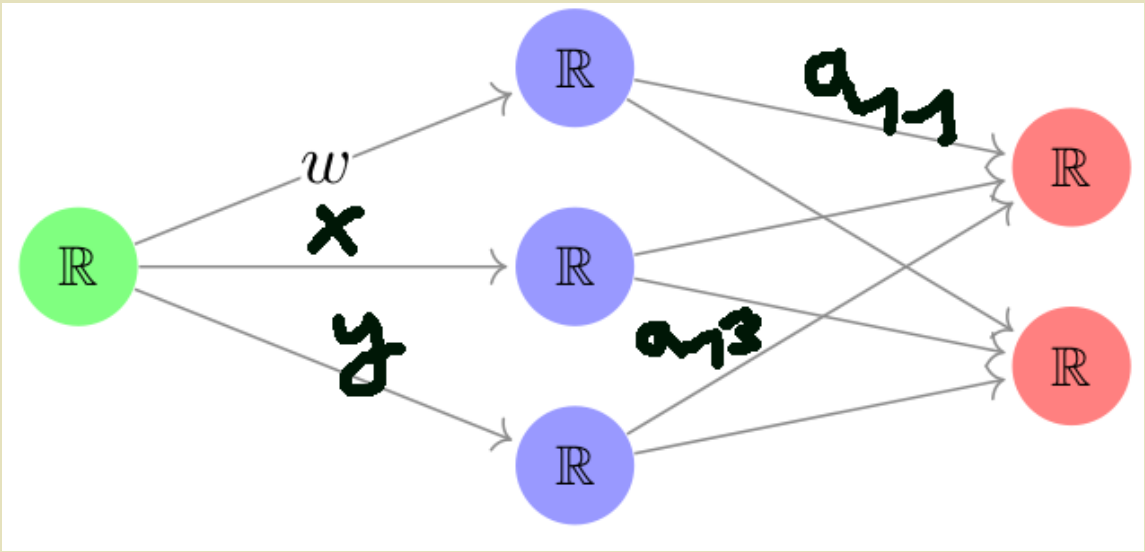



- NN = a directed graph as above
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

# Algebra in AI

**Example**




Here we have two matrices, a 3-by-1 and a 2-by-3 matrix

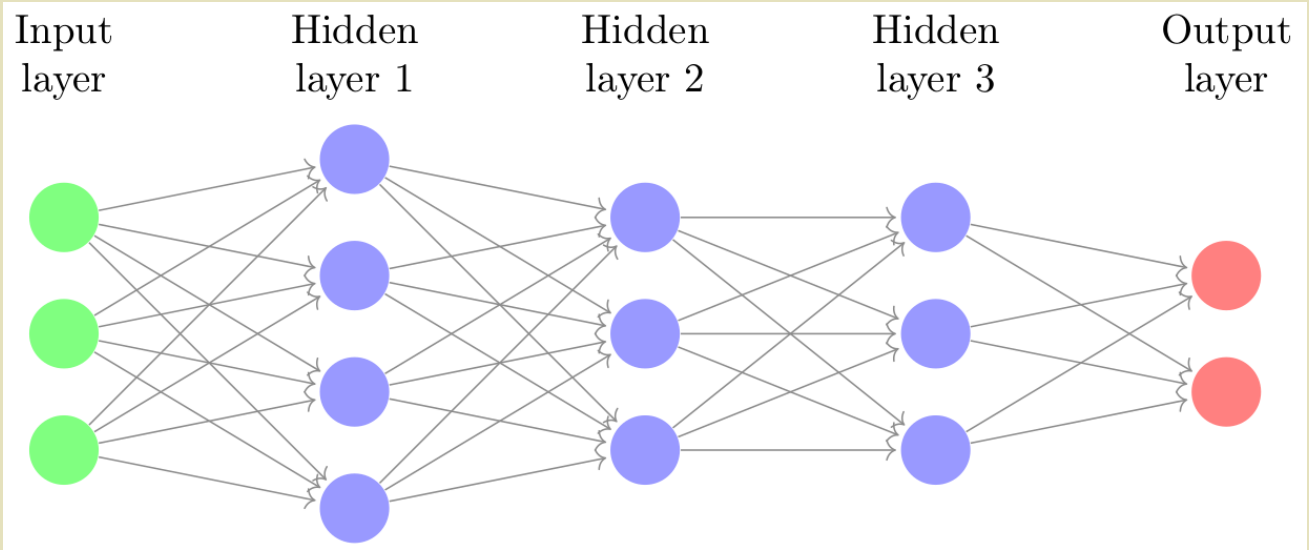
$$\begin{pmatrix} w \\ x \\ y \end{pmatrix} \text{ and } \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

plus two bias terms as in $y = matrix \cdot x + bias$

- NN = a
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

# Algebra in AI

**Example**




Here we have four matrices (plus four biases), whose composition gives a map

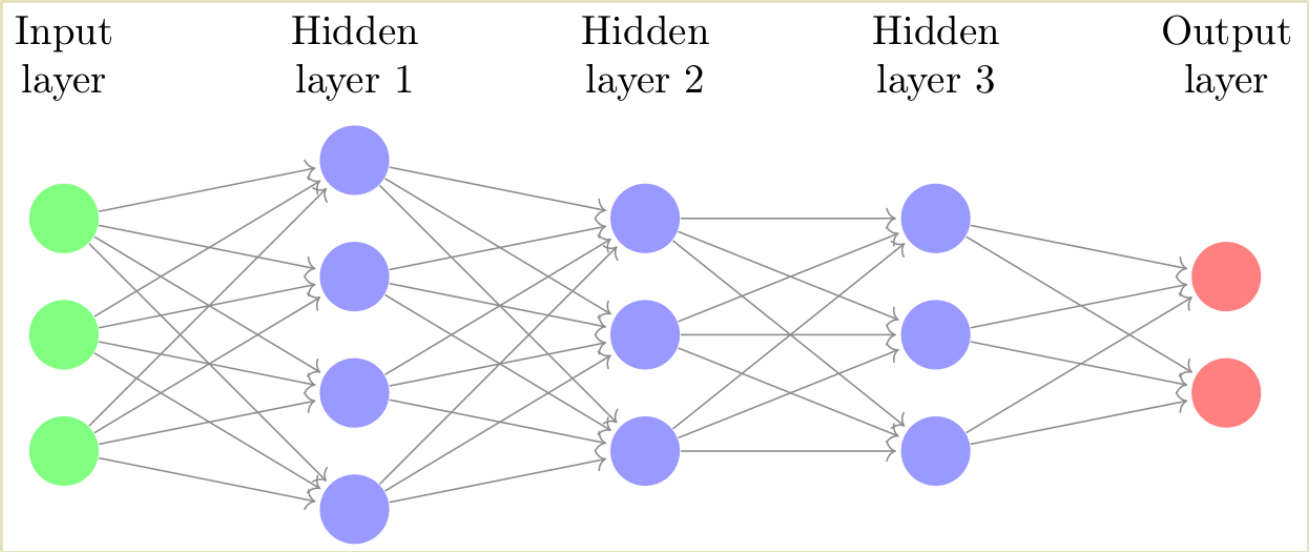
$$\mathbb{R}^3 \to \mathbb{R}^4 \to \mathbb{R}^3 \to \mathbb{R}^3 \to \mathbb{R}^2$$

- NN = a directed graph as above
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

# Algebra in AI

**Actually...**

we need nonlinear maps as well, say ReLU applied componentwise




Here we have four matrices, whose composition gives a map

$$\mathbb{R}^3 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^4 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^3 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^3 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^2$$

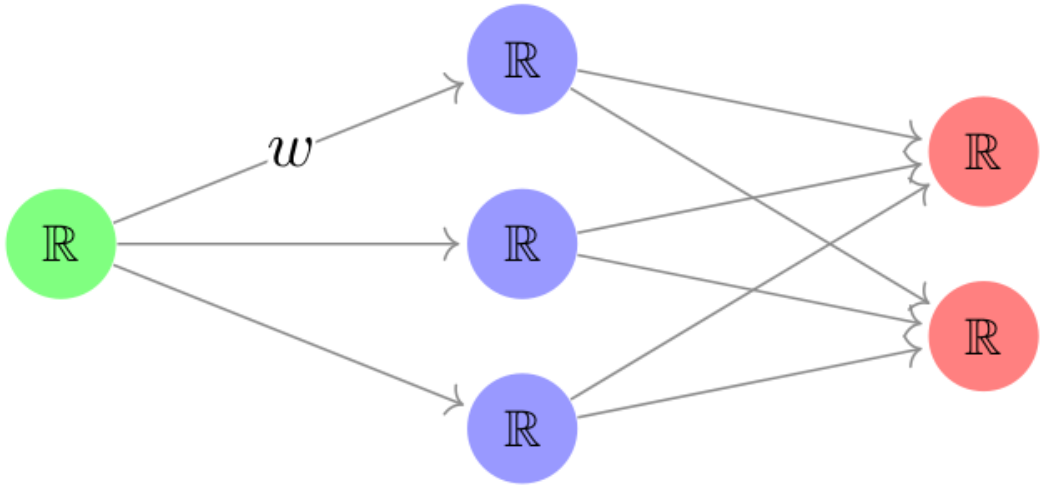
- NN = a directed graph as above
- The task

**But ignore that for now**

ReLU doesn't learn anything, its just brings in nonlinearity

- It consist of neurons = entries of vectors, and weights = entries of matrices

# Algebra in AI



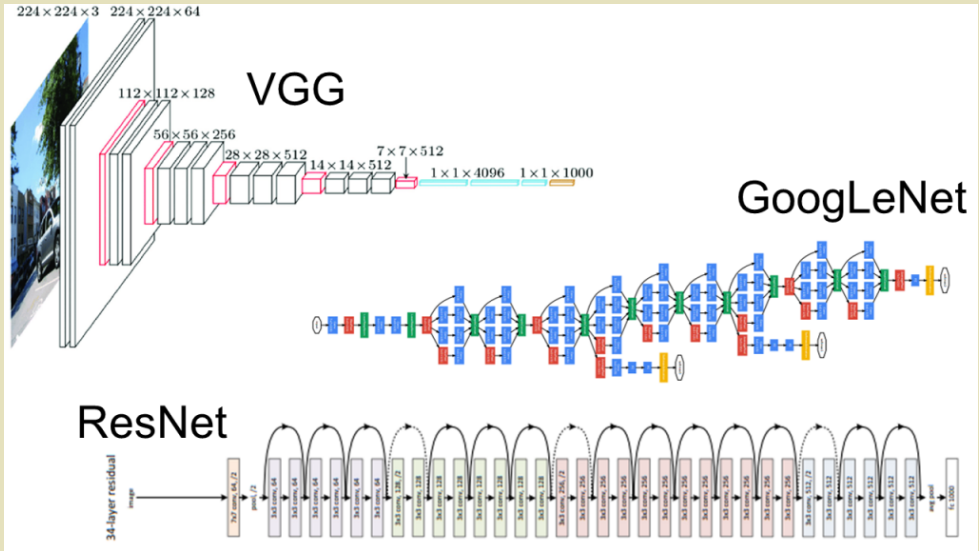

$$\left(\begin{array}{c|c} a^1_{11} & b^1_1 \\ a^1_{12} & b^1_2 \\ a^1_{13} & b^1_3 \end{array}\right), \quad \left(\begin{array}{ccc|c} a^2_{11} & a^2_{12} & a^2_{13} & b^2_1 \\ a^2_{21} & a^2_{22} & a^2_{23} & b^2_2 \end{array}\right)$$

- The $a^k_{ij}$ and $b^k_i$ are the parameters of our nn
- $k$ = number of the layer
- Deep = many layers = better approximation

**The point**

Many layers → many parameters
These are good for approximating real world problems

**Examples**



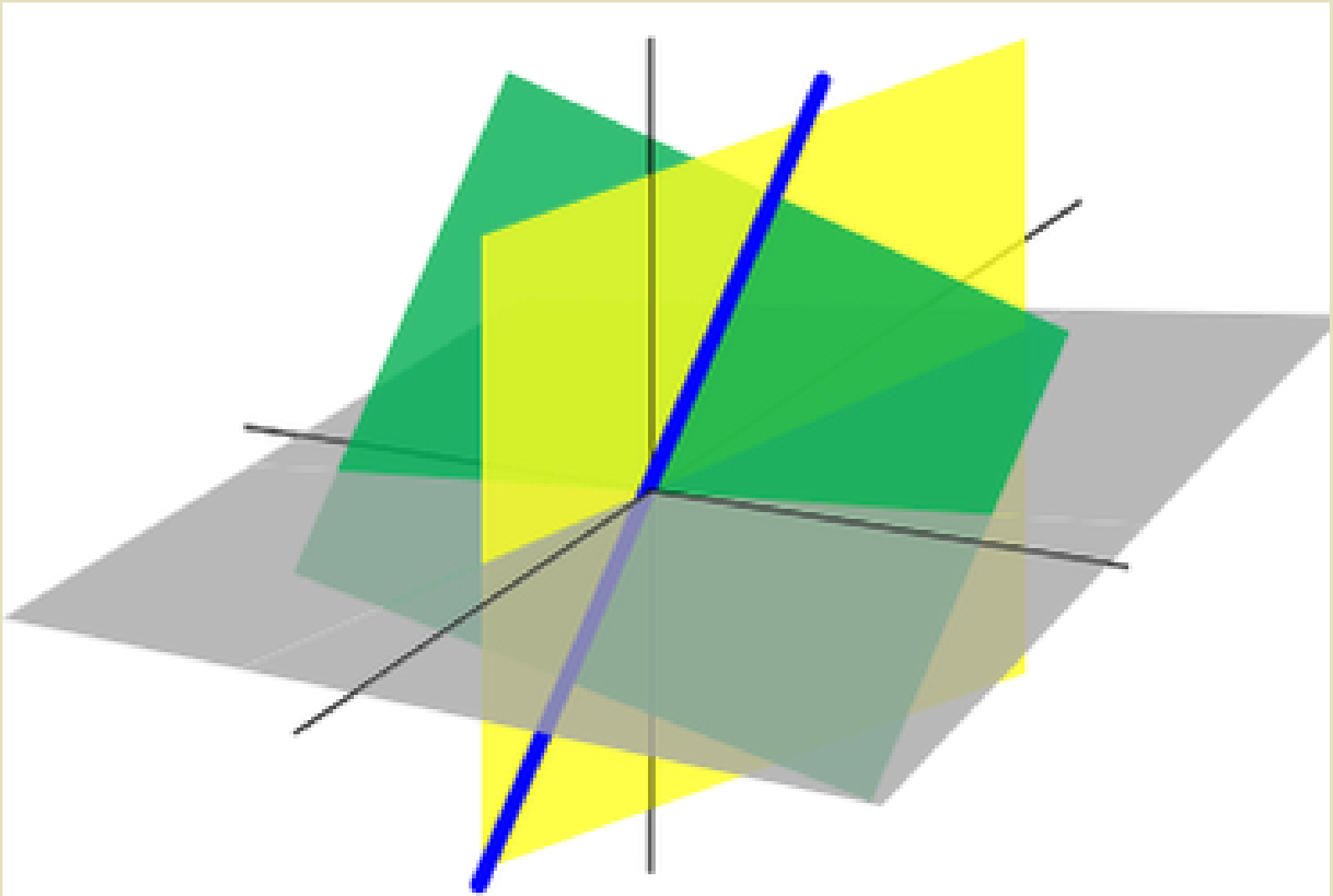

ResNet-152 with 152. layers (used in transformer models such as ChatGPT)
VGG-19 with 19 layers (used in image classification)
GoogLeNet with 22 layers (used in face detection)

- [illegible]
- $k$ = number of the layer
- Deep = many layers = better approximation

## Algeb

**In other words**

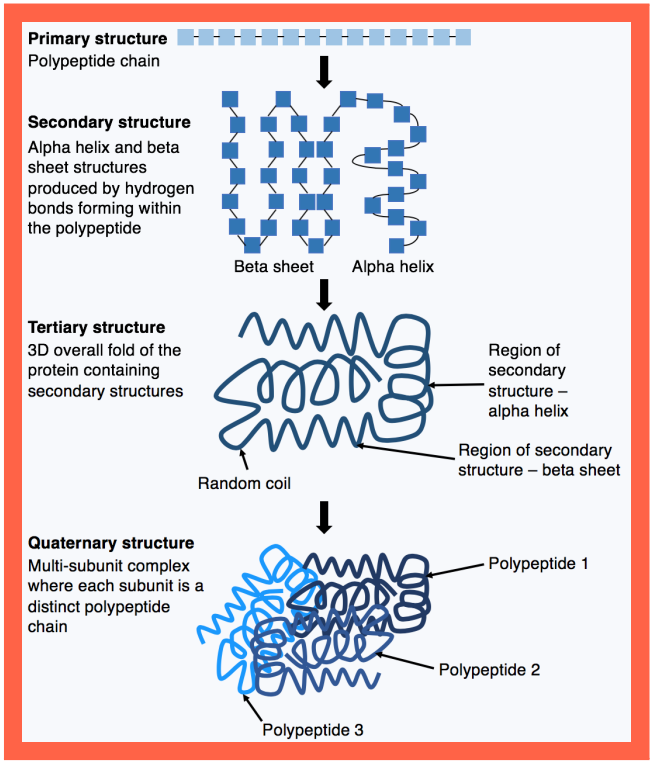
AI = matrix multiplication + some non-linear algebra



- T
- $k$
- D — many layers — better approximation

# AI in algebra

AI in science




- AlphaFold = AI model predicting protein folding
- It has transformed structural biology with accurate protein structure predictions
- This is one example of many ML breakthroughs in science, from drug discovery to climate models

# AI in algebra

General AI ChatGPT and friends are ok for most tasks (think: laptop)

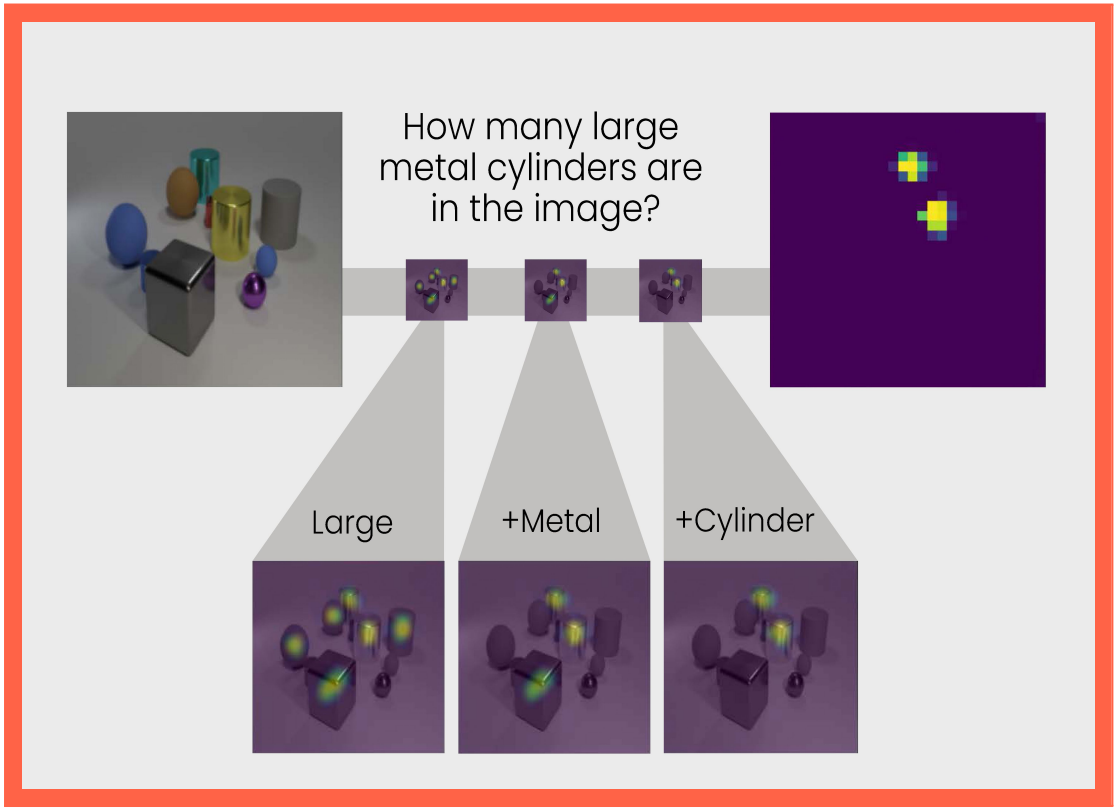
Specialized AI There are many models that are excellent for one task (think: compass)

Research tasks are mostly done by specialized AI

- It has transformed structural biology with accurate protein structure predictions
- This is one example of many ML breakthroughs in science, from drug discovery to climate models

# AI in algebra

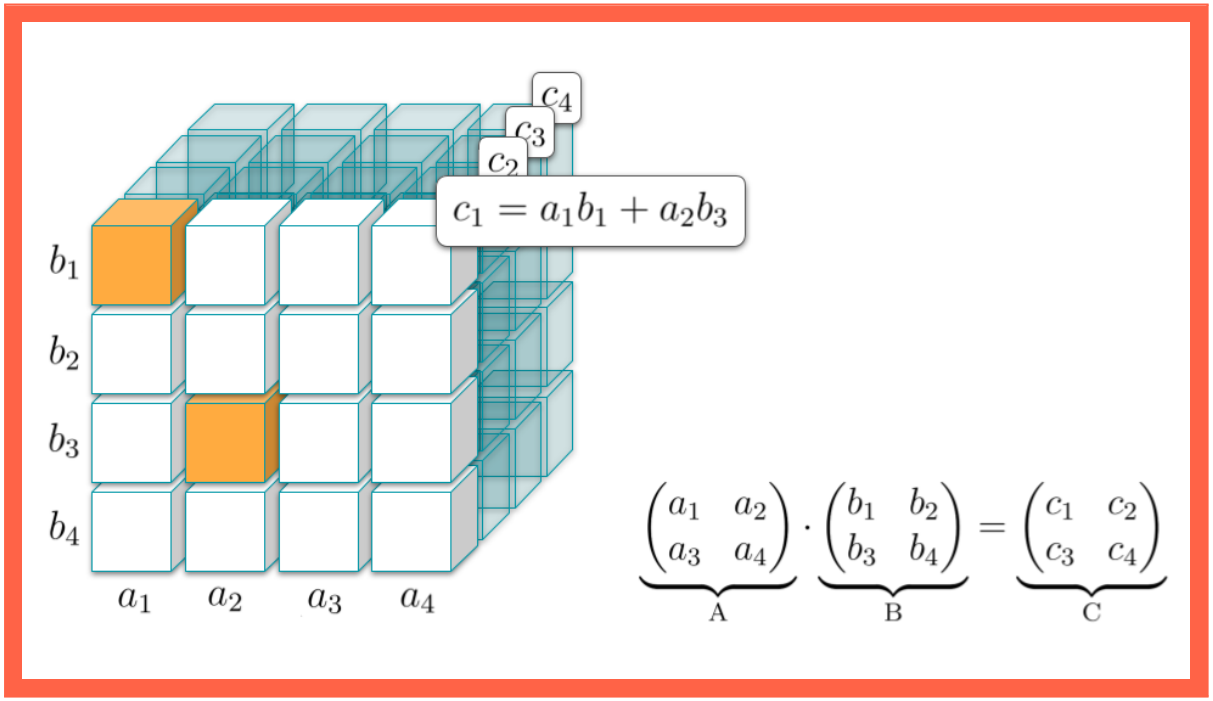
AI in math?




- Simulating reasoning is still one of the main open problems in ML
- Mathematics = the art of reasoning
- Problem That doesn't seem to fit together

# AI in algebra

AI in algebra




- AlphaTensor = specialized type of neural network that applies deep learning to algorithmic problem-solving
- It has found new algorithms to speed-up matrix multiplication (improved upon known methods)
- A main issue in ML is to have efficient matrix multiplication, so this is self-improving

## AI in algebra

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{pmatrix}$$

**Trick:**

$P_1 = A \cdot (F-H)$ $\quad$ $P_5 = (A+D) \cdot (E+H)$

$P_2 = (A+B) \cdot H$ $\quad$ $P_6 = (B-D) \cdot (G+H)$

$P_3 = (C+D) \cdot E$ $\quad$ $P_7 = (A-C) \cdot (E+F)$

$P_4 = D \cdot (G-E)$

$AE+BG = P_5 + P_4 - P_2 + P_6$

$AF+BH = P_1 + P_2$

$CE+DG = P_3 + P_4$

$CF+DH = P_5 + P_1 - P_3 - P_7$

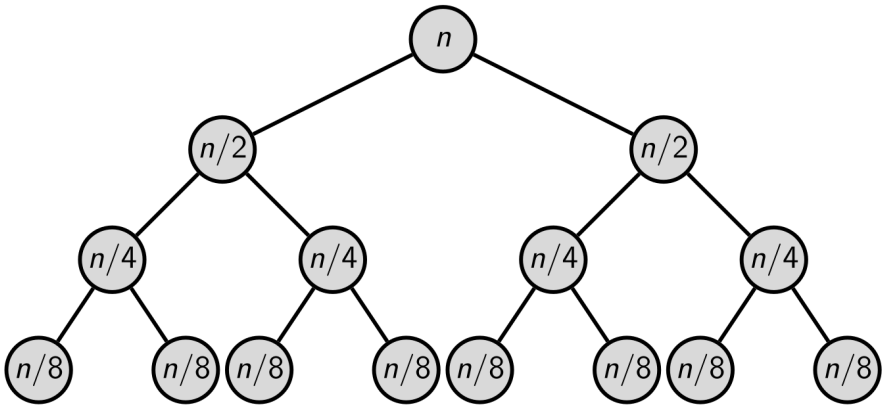
- Standard 2-by-2 matrix multiplication needs eight multiplications
- Strassen We only need seven
- Observation The seven multiplications are not trivial to find

# AI in algebra

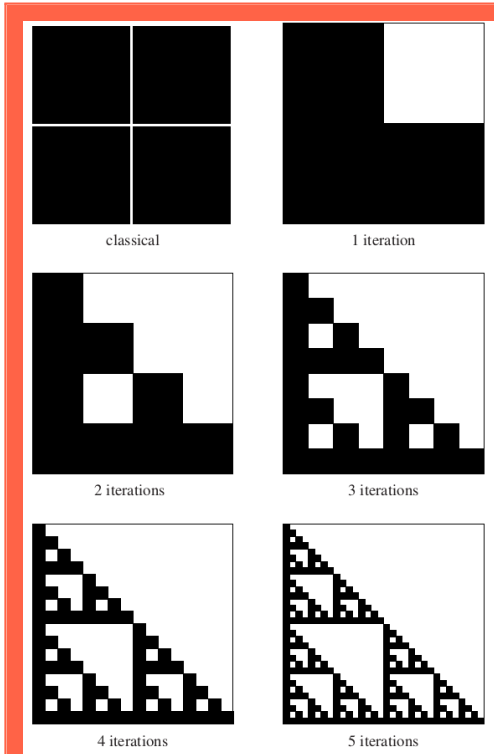
- Break the matrices in blocks of size $n/2 \times n/2$ **Divide!**

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} S1 + S4 - S5 + S7 & S3 + S5 \\ S2 + S4 & S1 - S2 + S3 + S6 \end{pmatrix}$$

- Needs 7 calls on $n/2 \times n/2$ matrices
- Repeat recursively **Conquer!**




Strassen then needs $\approx n^{\log_2(7)}$ operations




8.2: Cost (= black area) of Karatsuba's algorithm for increasing recursion depths.
ge approaches a fractal of dimension $\log_2 3 \approx 1.59$.

- **Recursion** Safe one operation each step while multiplying $2^n$-by-$2^n$ matrices
- **Standard multiplication** Needs $n^3$ operations $(3 = \log_2 8)$
- **Strassen multiplication** Needs $n^{2.81}$ operations $(2.81 \approx \log_2 7)$

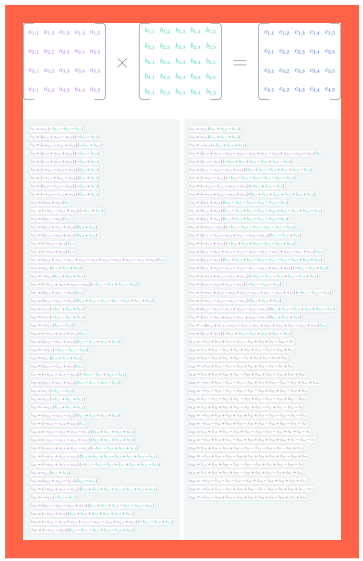
## AI in algebra

Deepmind's AlphaTensor NN found a 5-by-5 algorithm with 76 operations

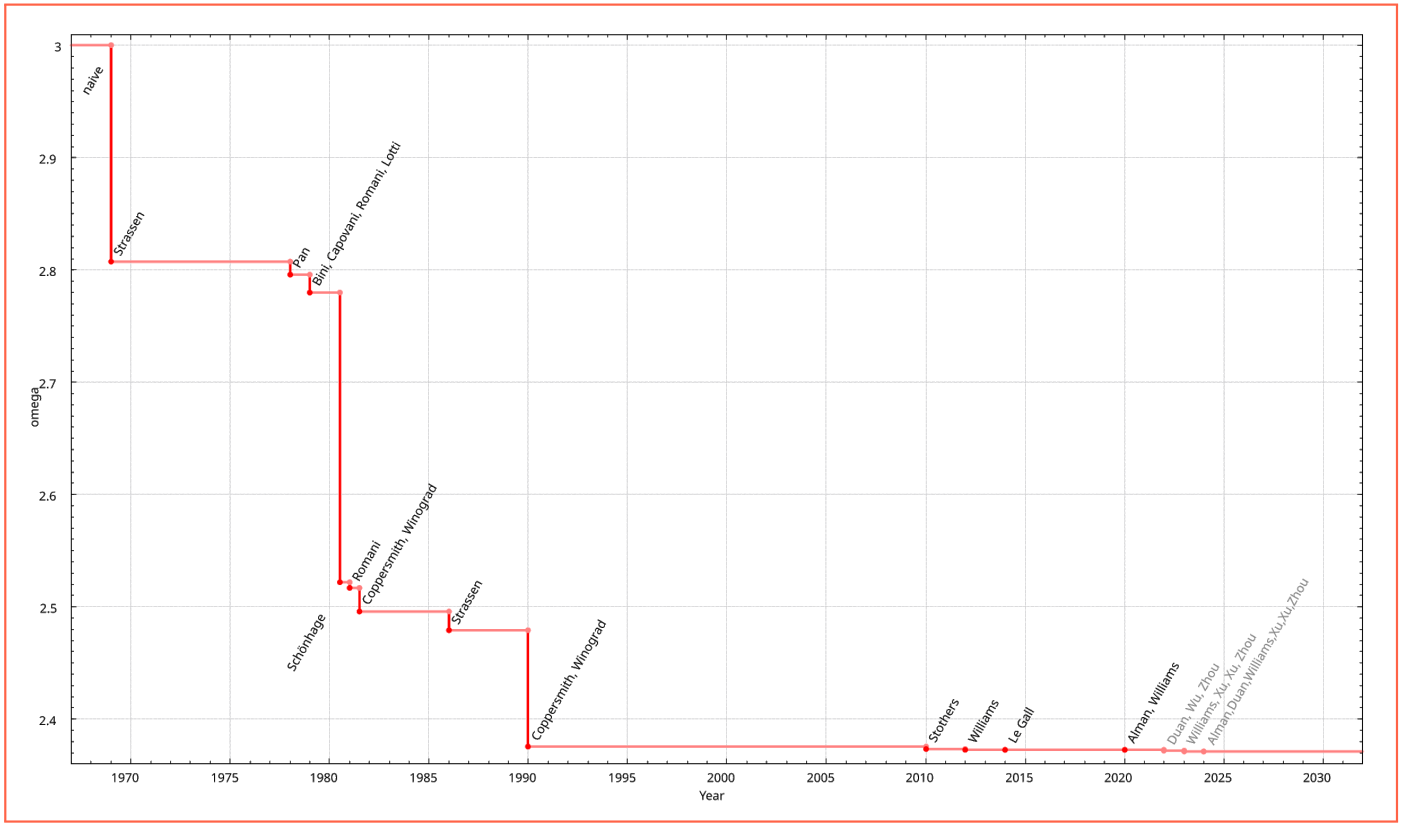
$$\log_5 76 \approx 2.69$$

This in turn makes NNs faster

- They actually discovered several other algorithms as well
- Here is the list Don't look ☺

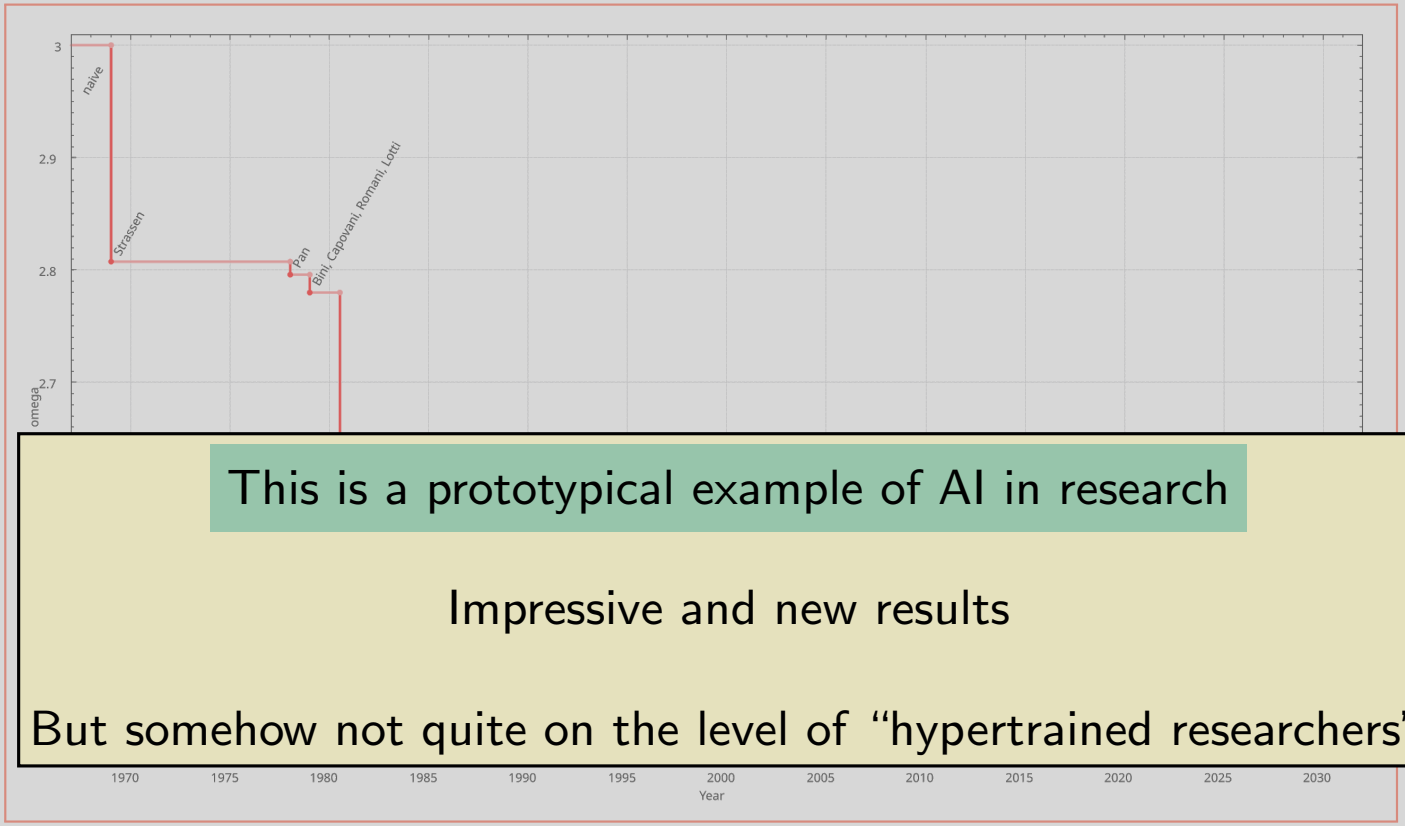


# AI in algebra




- There are actually faster algorithms
- Most of the newer ones use different methods
- Problem Most of them are not of practical importance

# AI in algebra




This is a prototypical example of AI in research

Impressive and new results

But somehow not quite on the level of "hypertrained researchers"

- There are actually faster algorithms
- Most of the newer ones use different methods
- Problem Most of them are not of practical importance

## AI in algebra

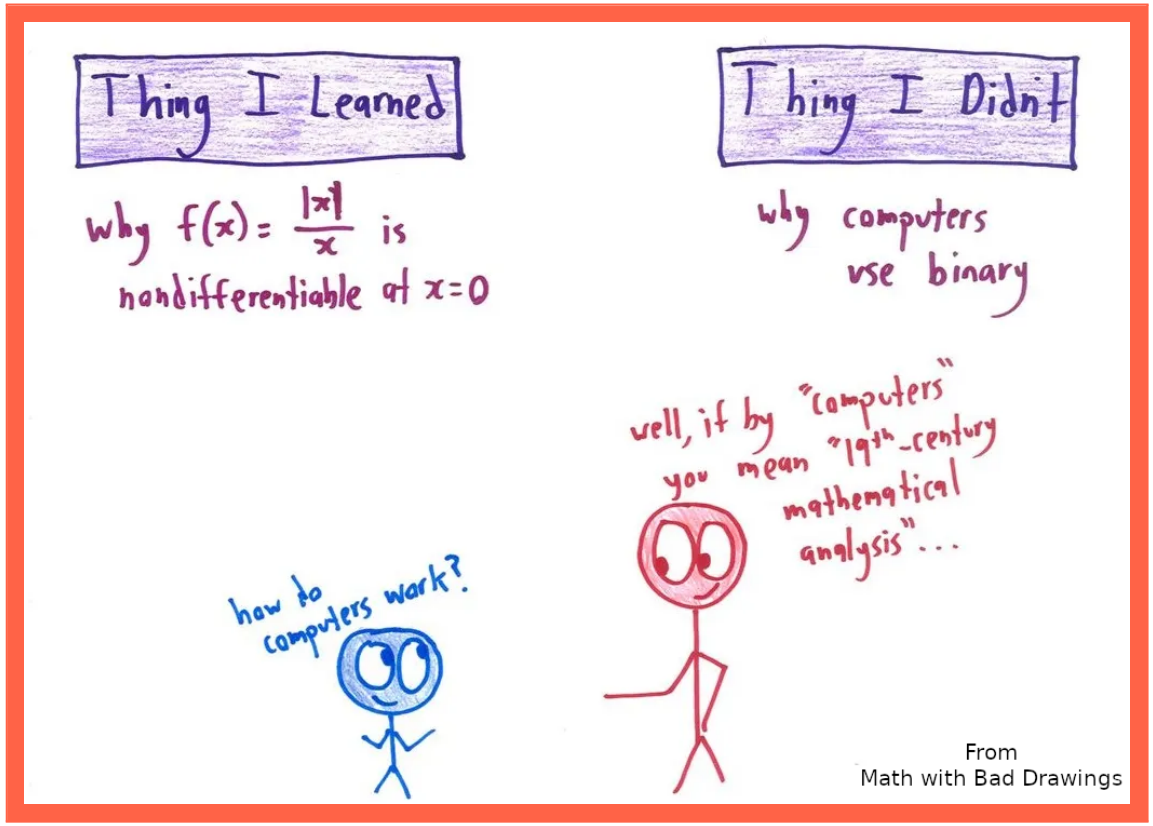
Examples of AI usage in algebra (and beyond)

- **Translation** Translate written into formal math (e.g. for proof verification)
- **Automated proofs** At the moment mostly closing proofs
- **Formula generation** E.g. find new formulas for "famous constants"
- **Approximating hard problems** E.g. finding Gröbner bases
- **Anit-conjecturing** Creating counterexamples
- **Pattern prediction** E.g. can one detect groups out of Latin squares

| $\cdot$ | $1$ | $x$ | $y$ | $z$ |
|---|---|---|---|---|
| $1$ | $1$ | $x$ | $y$ | $z$ |
| $x$ | $x$ | $y$ | $z$ | $1$ |
| $y$ | $y$ | $z$ | $1$ | $x$ |
| $z$ | $z$ | $1$ | $x$ | $y$ |

| $\cdot$ | $1$ | $x$ | $y$ | $z$ |
|---|---|---|---|---|
| $1$ | $1$ | $x$ | $y$ | $z$ |
| $x$ | $x$ | $1$ | $z$ | $y$ |
| $y$ | $y$ | $z$ | $1$ | $x$ |
| $z$ | $z$ | $y$ | $x$ | $1$ |

# AI in algebra



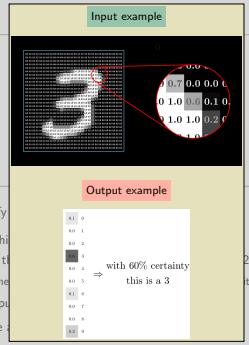

- Actually in mid 2025, ML in math is growing but not yet popular
- There are many reasons for that (some will be addressed in later videos)
- One is that mathematicians are not trained to work in ML

### Algebra in AI

- Task Identify handwritten digits
- We can see this as a function in the following way:
  - Convert the pictures into grayscale values, e.g. $28 \times 28$ grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$



### Algebra in AI

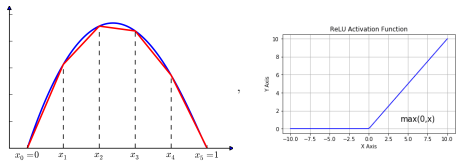



- Task Identify
- We can see thi
  - Convert t
  - Flatten the
  - The outpu
- We thus have

28 grid of numbers
th $28^2 = 784$ entries



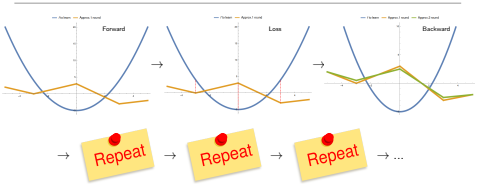
### Algebra in AI




- Idea Approximate the unknown function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
- Neural network = a piecewise linear approximation (matrices + PL maps)
- The matrices = a bunch of numbers (weights) and offsets (biases)
- The PL maps = usually ReLU



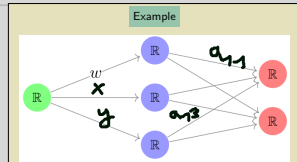
### Algebra in AI




- Machine learning mantra
  - Forward = calculate an approximation (start with random inputs)
  - Loss = compare to real data
  - Backward = adjust the approximation



### Algebra in AI




Here we have two matrices, a 3-by-1 and a 2-by-3 matrix

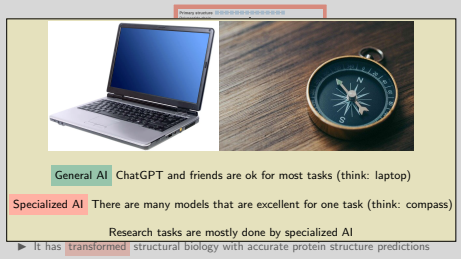
$$\begin{pmatrix} w \\ x \\ y \end{pmatrix} \text{ and } \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

plus two bias terms as in $y = matrix \cdot x + bias$

- NN = a
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices



### AI in algebra



General AI ChatGPT and friends are ok for most tasks (think: laptop)

Specialized AI There are many models that are excellent for one task (think: compass)

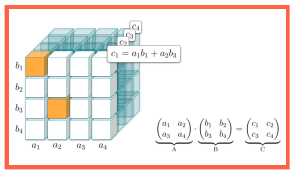
Research tasks are mostly done by specialized AI

- It has transformed structural biology with accurate protein structure predictions
- This is one example of many ML breakthroughs in science, from drug discovery to climate models



### AI in algebra

AI in algebra



- AlphaTensor = specialized type of neural network that applies deep learning to algorithmic problem-solving
- It has found new algorithms to speed-up matrix multiplication (improved upon known methods)
- A main issue in ML is to have efficient matrix multiplication, so this is self-improving



### AI in algebra

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{pmatrix}$$

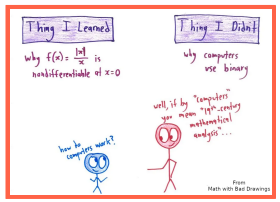
**Trick:**

| | | |
|---|---|---|
| $P_1 = A \cdot (F-H)$ | $P_5 = (A+D) \cdot (E+H)$ | $AE+BG = P_5 + P_4 - P_2 + P_6$ |
| $P_2 = (A+B) \cdot H$ | $P_6 = (B-D) \cdot (G+H)$ | $AF+BH = P_1 + P_2$ |
| $P_3 = (C+D) \cdot E$ | $P_7 = (A-C) \cdot (E+F)$ | $CE+DG = P_3 + P_4$ |
| $P_4 = D \cdot (G-E)$ | | $CF+DH = P_5 + P_1 - P_3 - P_7$ |

- Standard 2-by-2 matrix multiplication needs eight multiplications
- Strassen We only need seven
- Observation The seven multiplications are not trivial to find



### AI in algebra




- Actually in mid 2025, ML in math is growing but not yet popular
- There are many reasons for that (some will be addressed in later videos)
- One is that mathematicians are not trained to work in ML



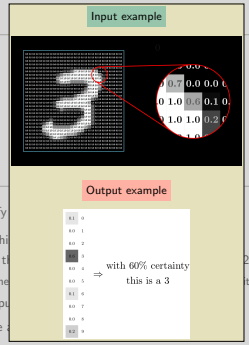
There is still much to do...

### Algebra in AI

- Task Identify handwritten digits
- We can see this as a function in the following way:
  - Convert the pictures into grayscale values, e.g. $28 \times 28$ grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$



### Algebra in AI

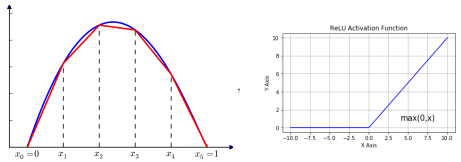



- Task Identify
- We can see thi
  - Convert t
  - Flatten the
  - The outpu
- We thus have

28 grid of numbers
th $28^2 = 784$ entries



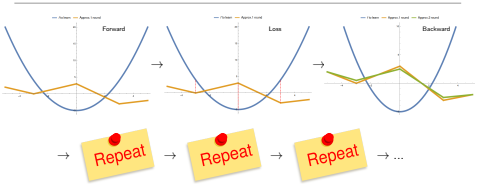
### Algebra in AI




- Idea Approximate the unknown function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
- Neural network = a piecewise linear approximation (matrices + PL maps)
- The matrices = a bunch of numbers (weights) and offsets (biases)
- The PL maps = usually ReLU



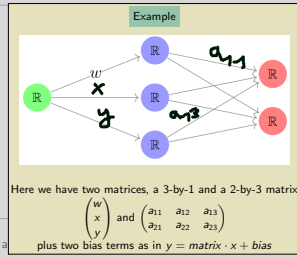
### Algebra in AI




- Machine learning mantra
  - Forward = calculate an approximation (start with random inputs)
  - Loss = compare to real data
  - Backward = adjust the approximation



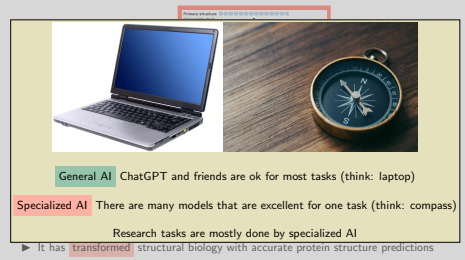
### Algebra in AI




- NN = a
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices



### AI in algebra



General AI ChatGPT and friends are ok for most tasks (think: laptop)

Specialized AI There are many models that are excellent for one task (think: compass)

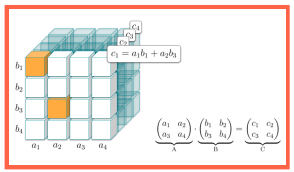
Research tasks are mostly done by specialized AI

- It has transformed structural biology with accurate protein structure predictions
- This is one example of many ML breakthroughs in science, from drug discovery to climate models



### AI in algebra

AI in algebra




- AlphaTensor = specialized type of neural network that applies deep learning to algorithmic problem-solving
- It has found new algorithms to speed-up matrix multiplication (improved upon known methods)
- A main issue in ML is to have efficient matrix multiplication, so this is self-improving



### AI in algebra

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{pmatrix}$$

**Trick:**

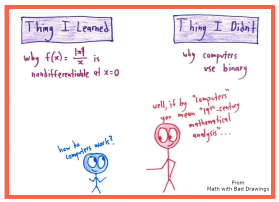
| | |
|---|---|
| $P_1 = A \cdot (F-H)$ | $P_5 = (A+D) \cdot (E+H)$ |
| $P_2 = (A+B) \cdot H$ | $P_6 = (B-D) \cdot (G+H)$ |
| $P_3 = (C+D) \cdot E$ | $P_7 = (A-C) \cdot (E+F)$ |
| $P_4 = D \cdot (G-E)$ | |

| |
|---|
| $AE+BG = P_5 + P_4 - P_2 + P_6$ |
| $AF+BH = P_1 + P_2$ |
| $CE+DG = P_3 + P_4$ |
| $CF+DH = P_5 + P_1 - P_3 - P_7$ |

- Standard 2-by-2 matrix multiplication needs eight multiplications
- Strassen We only need seven
- Observation The seven multiplications are not trivial to find



### AI in algebra




- Actually in mid 2025, ML in math is growing but not yet popular
- There are many reasons for that (some will be addressed in later videos)
- One is that mathematicians are not trained to work in ML



# Thanks for your attention!