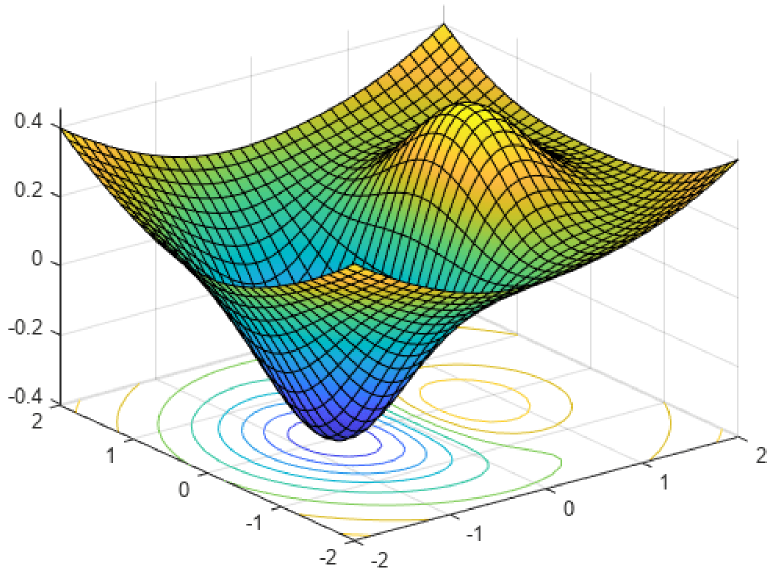
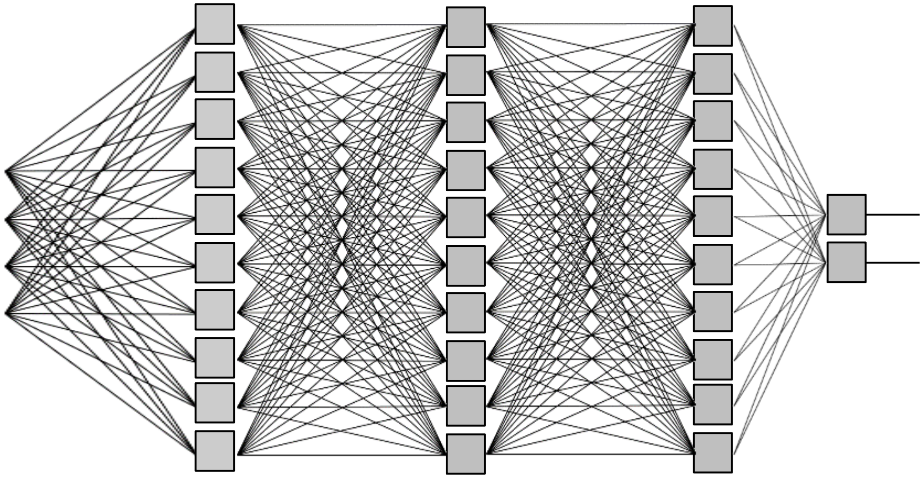
# Artificial Intelligence
# Machine Learning
# Deep Learning

Antoine Deleforge



# Organisation du module

- **Partie IA** (A. Deleforge):
  - 6 cours intégrés d'1h45:
    - 14/03am, 14/03pm, 27/03am, 27/03pm, 28/03pm, 12/04am
  - 3 TP de 4h:
    - Scindés en groupes A et B, du 29/03 jusqu'au 17/05

- **Partie Robotique** (L. Cuvillon):
  - 2 cours intégrés d'1h45:
    - 21/03am, 22/03am
  - 5 TP de 4h:
    - Scindés en groupes A et B, du 29/03 jusqu'au 17/05

# Organisation du module

- **Evaluation:**
  - Partie IA: Examen QCM + Compte rendus de TP
  - Partie Robotique: contrôle continu en TP

# Organisation du module

- **Evaluation:**
  - Partie IA: Examen QCM + Compte rendus de TP
  - Partie Robotique: contrôle continu en TP
- **Prérequis:**
  - Programmation python et orientée objet
  - Scalaires, vecteurs, matrices: $x, \boldsymbol{x}, \mathbf{X}$
  - Probas et statistiques: $p(x), p(x, y), p(x|y), \mathbb{E}, \mathrm{var}, \mathcal{N}$
  - Calcul différentiel: $\frac{\partial f}{\partial x}, \nabla f$
  - Optimisation

*Il y aura des rappels*

# Organisation du module

- **Evaluation:**
  - Partie IA: Examen QCM + Compte rendus de TP
  - Partie Robotique: contrôle continu en TP
- **Prérequis:**
  - Programmation python et orientée objet
  - Scalaires, vecteurs, matrices: $x, \boldsymbol{x}, \mathbf{X}$
  - Probas et statistiques: $p(x), p(x, y), p(x|y), \mathbb{E}, \mathrm{var}, \mathcal{N}$
  - Calcul différentiel: $\frac{\partial f}{\partial x}, \nabla f$
  - Optimisation

*Il y aura des rappels*

- Cette unité d'enseignement est nouvelle à TPS!

➡ Retours bienvenus et appréciés

➡ Soyez bienveillants et pro-actifs

# Sources

- **Ian Goodfellow**, **Yoshua Bengio**, **Aaron Courville**. *Deep Learning.* https://www.deeplearningbook.org/
- **Hugo Larochelle**, *Online Course on Neural Network.* http://info.usherbrooke.ca/hlarochelle/neural_networks/
- **Emmanuel Vincent**, *Neural Network* course. Master TAL, Univ. de Lorraine.
- **Paul Magron**, *Neural Network* labs. Master TAL, Université de Lorraine.
- **Antoine Liutkus**, cours *Deep Learning et réseaux de neurones, les fondamentaux.* Inria Sofia.
- https://towardsdatascience.com/
- https://cs230.stanford.edu/blog/pytorch/

# Mon Parcours

# Mon Parcours

**2007-2010:**

- Ecole d’ingénieur ENSIMAG (INPG)
- Double diplôme: master recherche graphisme, vision, robotique

# Mon Parcours

**2007-2010:**

- Ecole d’ingénieur ENSIMAG (INPG)
- Double diplôme: master recherche graphisme, vision, robotique

**2010-2013:**

- Thèse à l’Inria de Grenoble, équipe PERCEPTION (ajd: RobotLearn)
- Equipe en vision + machine learning / thèse en audio

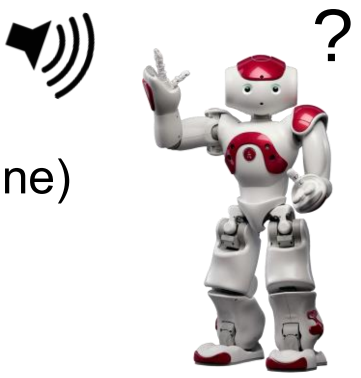
# Mon Parcours

**2007-2010:**

- Ecole d'ingénieur ENSIMAG (INPG)
- Double diplôme: master recherche graphisme, vision, robotique

**2010-2013:**

- Thèse à l'Inria de Grenoble, équipe PERCEPTION (ajd: RobotLearn)
- Equipe en vision + machine learning / thèse en audio




**2014-2015:**

- Post-doc à l'université Friedrich-Alexander d'Erlangen (Allemagne)
- Projet européen EARS sur l'Audition Robotique.

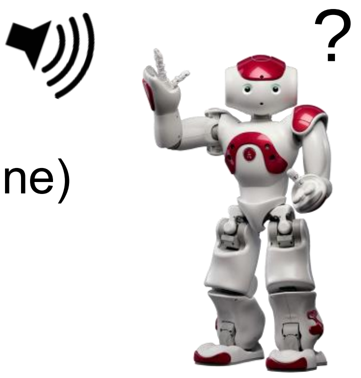
# Mon Parcours

**2007-2010:**

- Ecole d’ingénieur ENSIMAG (INPG)
- Double diplôme: master recherche graphisme, vision, robotique

**2010-2013:**

- Thèse à l’Inria de Grenoble, équipe PERCEPTION (ajd: RobotLearn)
- Equipe en vision + machine learning / thèse en audio




**2014-2015:**

- Post-doc à l’université Friedrich-Alexander d’Erlangen (Allemagne)
- Projet européen EARS sur l’Audition Robotique.

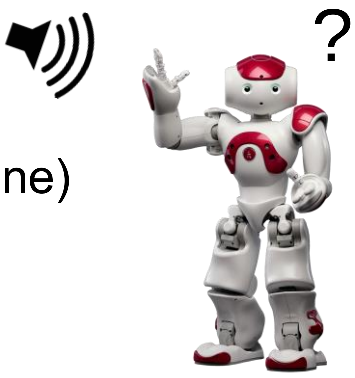
**2016-présent:**

- Chargé de recherche Inria
- Equipe PANAMA (Rennes) puis MULTISPEECH (Nancy) puis (bientôt) MACARON (Strasbourg)
- Ré-orientation vers l’acoustique des salles

# Mon Parcours

**2007-2010:**
- Ecole d’ingénieur ENSIMAG (INPG)
- Double diplôme: master recherche graphisme, vision, robotique

**2010-2013:**
- Thèse à l’Inria de Grenoble, équipe PERCEPTION (ajd: RobotLearn)
- Equipe en vision + machine learning / thèse en audio




**2014-2015:**
- Post-doc à l’université Friedrich-Alexander d’Erlangen (Allemagne)
- Projet européen EARS sur l’Audition Robotique.

**2016-présent:**
- Chargé de recherche Inria
- Equipe PANAMA (Rennes) puis MULTISPEECH (Nancy) puis (bientôt) MACARON (Strasbourg)
- Ré-orientation vers l’acoustique des salles

## Et vous?

# Concept du Cours

**Constats:**

- Essor des frameworks opensource depuis 2016 (TensorFlow, Pytorch,…)
  $\Longrightarrow$ Coder un algo de Deep Learning est devenu très accessible, en quelques tutos

- Un nouveau papier IA sort toutes les heures sur ArXiv, une nouvelle “révolution” toutes les semaines

# Concept du Cours

**Constats:**

- Essor des frameworks opensource depuis 2016 (TensorFlow, Pytorch,…)
  $\Rightarrow$ Coder un algo de Deep Learning est devenu très accessible, en quelques tutos

- Un nouveau papier IA sort toutes les heures sur ArXiv, une nouvelle "révolution" toutes les semaines

Se focaliser sur les **concepts fondamentaux** pour:

- Savoir **trier** le bon grain de l'ivraie
- Assimiler rapidement de **nouvelles** architectures et méthodes
- Identifier la meilleure approche pour un **cas d'usage**
- Acquérir des **bonnes pratiques**
- **Diagnostiquer** les problèmes

- Diapos en **anglais**

# OUTLINE

# OUTLINE

# OUTLINE

I. **Introduction**
- Artificial Intelligence
- Machine Learning
- Neural Network and Deep Learning
- Applications

II. Background

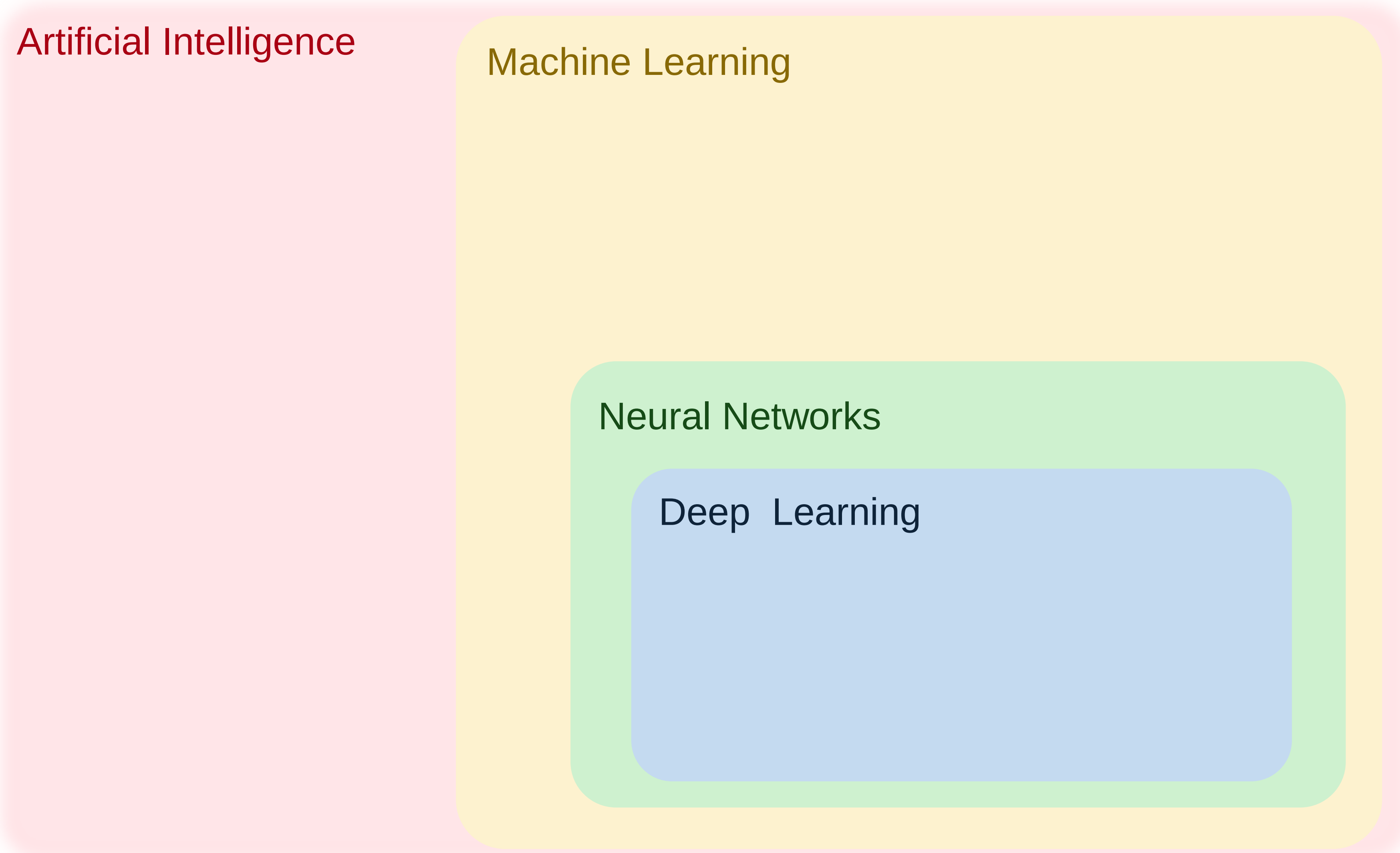
III. Fitting a Model

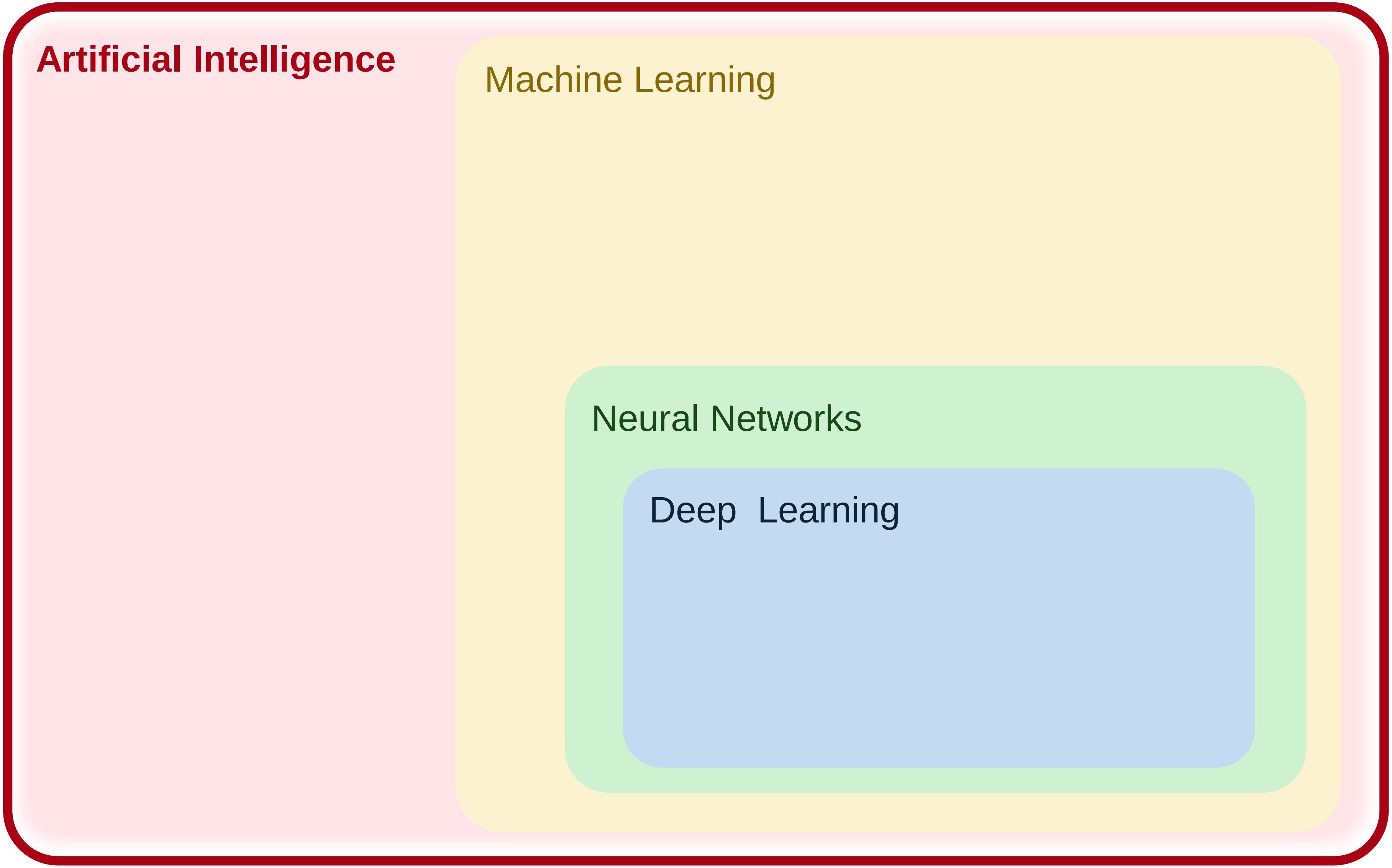
IV. Supervised Learning

V. Unsupervised Learning

VI. Fantastic DNNs: How to choose them, how to train them

VII. Machine Learning in Robot Audition


Artificial Intelligence
Machine Learning
Neural Networks
Deep Learning


Artificial Intelligence
Machine Learning
Neural Networks
Deep Learning

# What is Intelligence?

# What is Intelligence?

- A difficult question, no consensus today

## What is Intelligence?

- A difficult question, no consensus today

"A Collection of Definitions of Intelligence“, *Shane Legg, Marcus Hutter, 2007*. Frontiers in Artificial Intelligence and applications.

➡ 70 definitions!

# What is Intelligence?

- A difficult question, no consensus today

"A Collection of Definitions of Intelligence", *Shane Legg, Marcus Hutter, 2007*. Frontiers in Artificial Intelligence and applications.

70 definitions!

- Ex. of **dictionary** definition

- "The ability to use memory, knowledge, experience, understanding, reasoning, imagination and judgement in order to solve problems and adapt to new situations." *AllWords Dictionary, 2006*

# What is Intelligence?

- A difficult question, no consensus today

"A Collection of Definitions of Intelligence“, *Shane Legg, Marcus Hutter, 2007*. Frontiers in Artificial Intelligence and applications.

➡ 70 definitions!

- Ex. of **dictionary** definition

- "The ability to use memory, knowledge, experience, understanding, reasoning, imagination and judgement in order to solve problems and adapt to new situations." *AllWords Dictionary, 2006*

- Ex. of **psychologist** definitions

- "Intelligence is what is measured by intelligence tests." *E. Boring, 1923*

# What is Intelligence?

- A difficult question, no consensus today

  "A Collection of Definitions of Intelligence“, *Shane Legg, Marcus Hutter, 2007*. Frontiers in Artificial Intelligence and applications.

  → 70 definitions!

- Ex. of **dictionary** definition

  - "The ability to use memory, knowledge, experience, understanding, reasoning, imagination and judgement in order to solve problems and adapt to new situations." *AllWords Dictionary, 2006*

- Ex. of **psychologist** definitions

  - "Intelligence is what is measured by intelligence tests." *E. Boring, 1923*
  - "**Fluid** intelligence is your ability to process new information, learn, and solve problems. **Crystallized** intelligence is your stored knowledge, accumulated over the years." *R. Cattell, 1963*

# What is Intelligence?

- A difficult question, no consensus today

  "A Collection of Definitions of Intelligence“, *Shane Legg, Marcus Hutter, 2007*. Frontiers in Artificial Intelligence and applications.

  ➡ 70 definitions!

- Ex. of **dictionary** definition

  - "The ability to use memory, knowledge, experience, understanding, reasoning, imagination and judgement in order to solve problems and adapt to new situations." *AllWords Dictionary, 2006*

- Ex. of **psychologist** definitions

  - "Intelligence is what is measured by intelligence tests." *E. Boring, 1923*
  - "**Fluid** intelligence is your ability to process new information, learn, and solve problems. **Crystallized** intelligence is your stored knowledge, accumulated over the years." *R. Cattell, 1963*
  - "Intelligence is not a single, unitary ability, but rather a composite of several functions. The term denotes that combination of abilities required for survival and advancement within a particular culture." *A. Anastasi, 1992*

# What is Intelligence?

- Ex. of **AI researchers** definitions
  - "Intelligence is the ability to use optimally limited resources –including time– to achieve goals." *R. Kurzweil, 2000*

# What is Intelligence?

- Ex. of **AI researchers** definitions
  - "Intelligence is the ability to use optimally limited resources –including time– to achieve goals." *R. Kurzweil, 2000*
  - "Intelligence measures an agent's ability to achieve goals in a wide range of environments." *Shane Legg, Marcus Hutter, 2007*

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction


Talos protecting Europa in Cretes,
Greek Mythology, c. 400 BC

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction


Talos protecting Europa in Cretes,
Greek Mythology, c. 400 BC


Theater play R.U.R,
Karel Čapek, 1920

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction

Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction


Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968


M3GAN, Gerard Johnstone, 2022

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction


Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968


M3GAN, Gerard Johnstone, 2022

- Strongly embedded in **collective imagination**

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction


Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968


M3GAN, Gerard Johnstone, 2022

- Strongly embedded in **collective imagination**
- A **relative** notion

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction

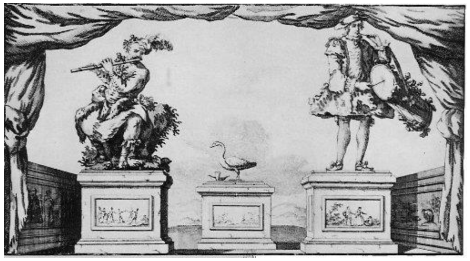
Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968

M3GAN, Gerard Johnstone, 2022

- Strongly embedded in **collective imagination**
- A **relative** notion



Vaucanson, 1737

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction

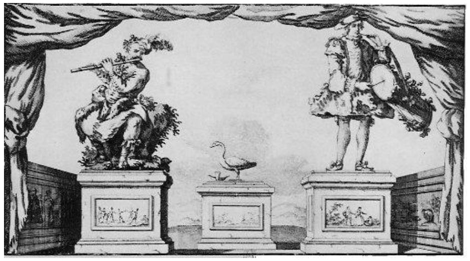

Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968


M3GAN, Gerard Johnstone, 2022

- Strongly embedded in **collective imagination**
- A **relative** notion



Vaucanson, 1737

Kasparov – Deep Blue 1997

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction

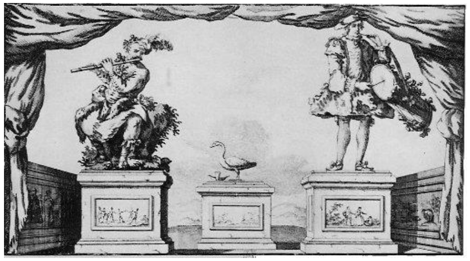
Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968

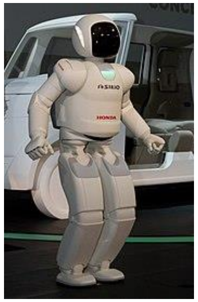
M3GAN, Gerard Johnstone, 2022

- Strongly embedded in **collective imagination**
- A **relative** notion



Vaucanson, 1737

Kasparov – Deep Blue 1997



ASIMO 2000

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction

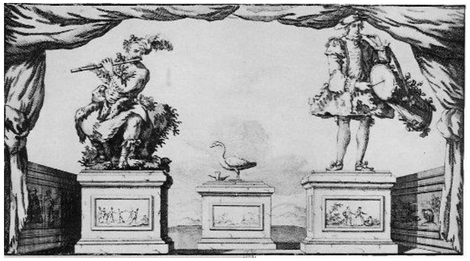
Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968

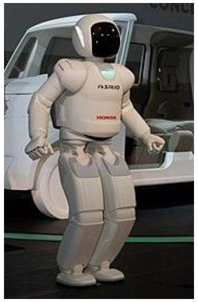

M3GAN, Gerard Johnstone, 2022

- Strongly embedded in **collective imagination**
- A **relative** notion



Vaucanson, 1737

Kasparov – Deep Blue 1997



ASIMO 2000


Lee Sedol – AlphaGo 2016

# Artificial Intelligence: an ill-defined term

- Originates in story telling, science fiction

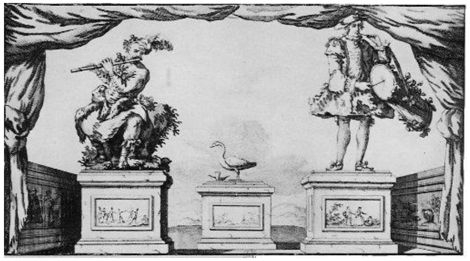
Talos protecting Europa in Cretes, Greek Mythology, c. 400 BC


Theater play R.U.R, Karel Čapek, 1920


HAL9000, 2001 Space Odyssee, Stanley Kubrik, 1968

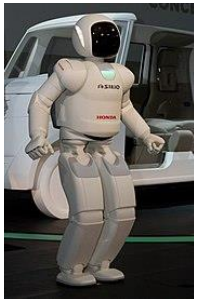
M3GAN, Gerard Johnstone, 2022

- Strongly embedded in **collective imagination**
- A **relative** notion



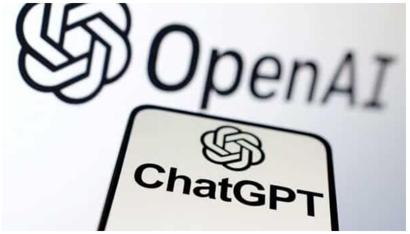
Vaucanson, 1737

Kasparov – Deep Blue 1997



ASIMO 2000


Lee Sedol – AlphaGo 2016



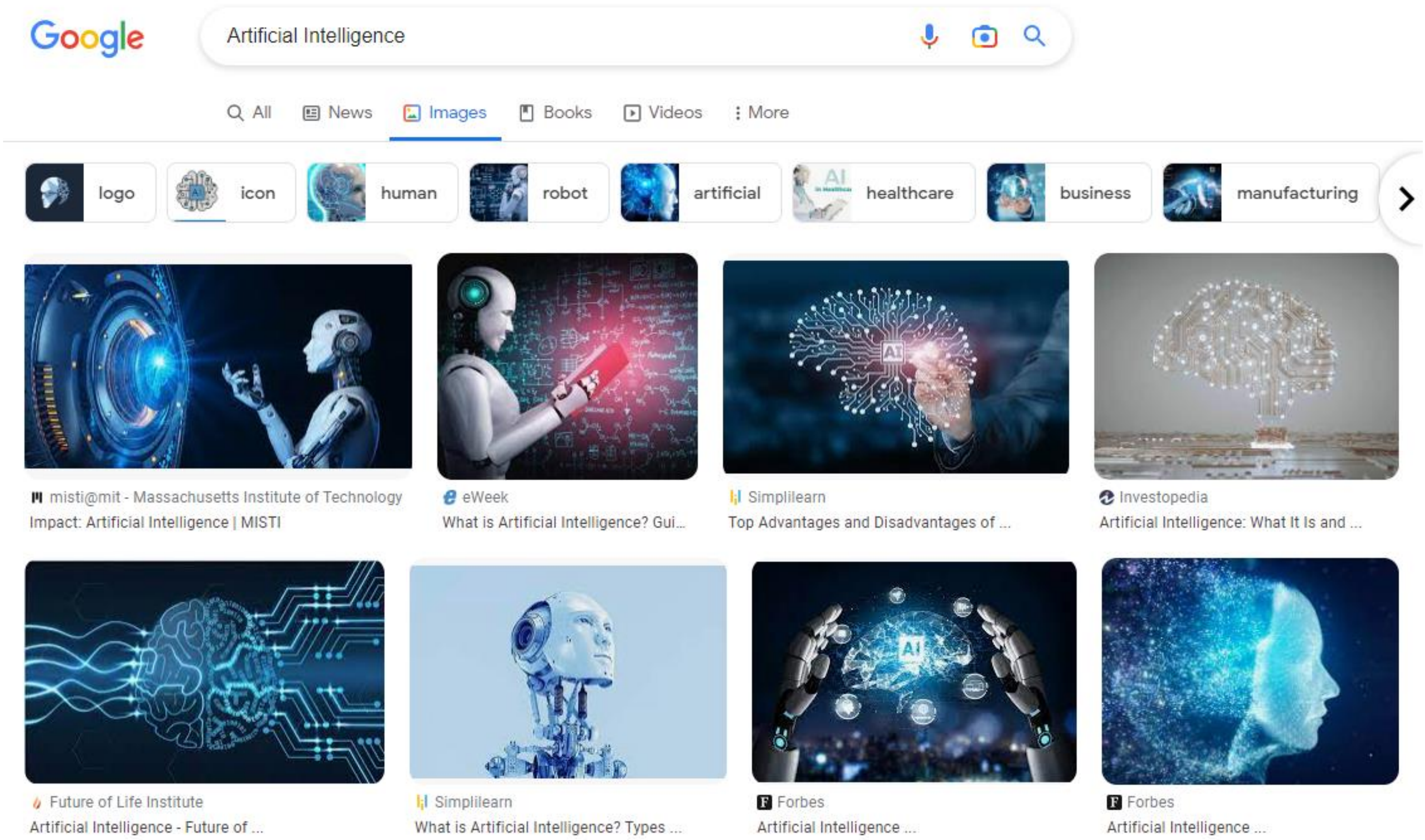

ChatGPT 2022

# Artificial Intelligence: an ill-defined term

- A common confusion: **A.I. == ROBOTICS**

# Artificial Intelligence: an ill-defined term

- A common confusion: **A.I. == ROBOTICS**



# Artificial Intelligence: an ill-defined term

- A common confusion: **A.I. == ROBOTICS**
- Human vs. Machine intelligence:
  - Highly Physical Tasks + Big Compute = Machines 🤖
  - Highly Creative and Intellectual Tasks = Humans 🧠

# Artificial Intelligence: an ill-defined term

- A common confusion: **A.I. == ROBOTICS**
- Human vs. Machine intelligence:
  - Highly Physical Tasks + Big Compute = Machines 🤖
  - Highly Creative and Intellectual Tasks = Humans 🧠

**Status in 2023**

- Have you seen a robot tidying up your appartment?
- Have you seen a machine:
  - Win an art contest? (2022)
  - Self-learn to play and beat humans at arbitrary games? (2020)
  - Hold coherent extended conversations? (2022)
  - Rewrite Bohemian Rhapsody, but about a post-doc’s life? (2020)

# Artificial Intelligence: an ill-defined term

- A.I. is a « catch-all » word

# Artificial Intelligence: an ill-defined term

- A.I. is a « catch-all » word
- Rarely used in scientific publications

*Occurrence of terms in 12,900* ***conference paper titles*** *published at "Neural Information Processing Systems" since 2010* *[Source: Google Scholar]*

```
Learning: 3,310
Neural: 1,260
Deep: 864
Deep Learning: 291
Neural Network: 151
Machine Learning: 107
Artificial: 10
Intelligence: 3
Artificial Intelligence: 1
```

# Artificial Intelligence: an ill-defined term

- A.I. is a « catch-all » word
- Rarely used in scientific publications

  *Occurrence of terms in 12,900* ***conference paper titles*** *published at "Neural Information Processing Systems" since 2010* *[Source: Google Scholar]*

```
Learning: 3,310
Neural: 1,260
Deep: 864
Deep Learning: 291
Neural Network: 151
Machine Learning: 107
Artificial: 10
Intelligence: 3
Artificial Intelligence: 1
```

- Ex: Is signal processing / statistics / optimization A.I.?

# Artificial Intelligence: an ill-defined term

- A.I. is a « catch-all » word
- Rarely used in scientific publications

*Occurrence of terms in 12,900 **conference paper titles** published at "Neural Information Processing Systems" since 2010* [Source: Google Scholar]

```
Learning: 3,310
Neural: 1,260
Deep: 864
Deep Learning: 291
Neural Network: 151
Machine Learning: 107
Artificial: 10
Intelligence: 3
Artificial Intelligence: 1
```

- Ex: Is signal processing / statistics / optimization A.I.?
- Understood by the general public = good for science communication

# Artificial Intelligence: an ill-defined term

- A.I. is a « catch-all » word
- Rarely used in scientific publications

*Occurrence of terms in 12,900 **conference paper titles** published at "Neural Information Processing Systems" since 2010 [Source: Google Scholar]*

```
Learning: 3,310
Neural: 1,260
Deep: 864
Deep Learning: 291
Neural Network: 151
Machine Learning: 107
Artificial: 10
Intelligence: 3
Artificial Intelligence: 1
```

Source: DALL-E, openai.com

- Ex: Is signal processing / statistics / optimization A.I.?
- Understood by the general public = good for science communication
- Understood by decision makers = good for getting funding

# The Rise of A.I.

- Over the past 10 years (~2012), an **explosion** of the term A.I.
- Mostly in **media headlines** and for **marketing** purposes ($$)
- Beware of publicity stunts / Wizards of Oz

# The Rise of A.I.

- Over the past 10 years (~2012), an **explosion** of the term A.I.

- Mostly in **media headlines** and for **marketing** purposes ($$)

- Beware of publicity stunts / Wizards of Oz


**Engineer.ia:** An Indian start-up that raised 30M€ by claiming to use "human-assisted AI tools" to develop mobile apps in record time. Following a lawsuit from employees, it was revealed that AI was mostly used as a marketing term (2019).

# The Rise of A.I.

- Over the past 10 years (~2012), an **explosion** of the term A.I.
- Mostly in **media headlines** and for **marketing** purposes ($$)
- Beware of publicity stunts / Wizards of Oz


**Engineer.ia:** An Indian start-up that raised 30M€ by claiming to use "human-assisted AI tools" to develop mobile apps in record time. Following a lawsuit from employees, it was revealed that AI was mostly used as a marketing term (2019).

**Pinscreen:** same idea but for generating 3D avatars from photos (2018)

# The Rise of A.I.

- Over the past 10 years (~2012), an **explosion** of the term A.I.
- Mostly in **media headlines** and for **marketing** purposes ($$)
- Beware of publicity stunts / Wizards of Oz


**Engineer.ia:** An Indian start-up that raised 30M€ by claiming to use "human-assisted AI tools" to develop mobile apps in record time. Following a lawsuit from employees, it was revealed that AI was mostly used as a marketing term (2019).

engineer.ai

**Pinscreen:** same idea but for generating 3D avatars from photos (2018)

**Sophia:** The first humanoid robot "acquiring citizenship", from Saudi Arabia (2017)

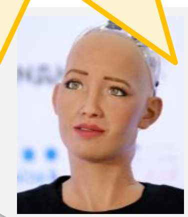
# The Rise of A.I.

- Over the past 10 years (~2012), an **explosion** of the term A.I.
- Mostly in **media headlines** and for **marketing** purposes ($$)
- Beware of publicity stunts / Wizards of Oz


**Engineer.ia:** An Indian start- … that r… ed 30M … by claiming to use “human-assisted … …bile apps in record time. Following a … it was revealed that AI was mostly us… …19).

So is A.I. just a scam?!

**Pinscreen:** same idea but for generating 3D avatars from photos (2018)



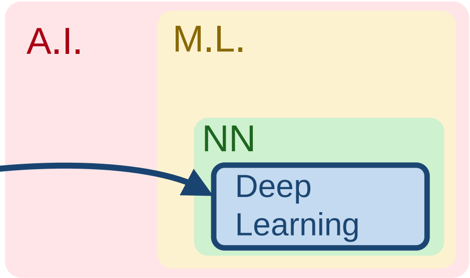
**Sophia:** The first humanoid robot “acquiring citizenship”, from Saudi Arabia (2017)

# The Rise of A.I.

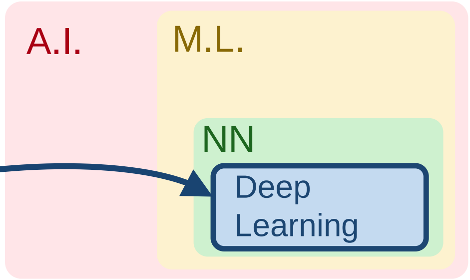
- **No.** A real **revolution** is taking shape

# The Rise of A.I.

- **No.** A real **revolution** is taking shape
- The core driver is not « AI » but **Deep Learning**

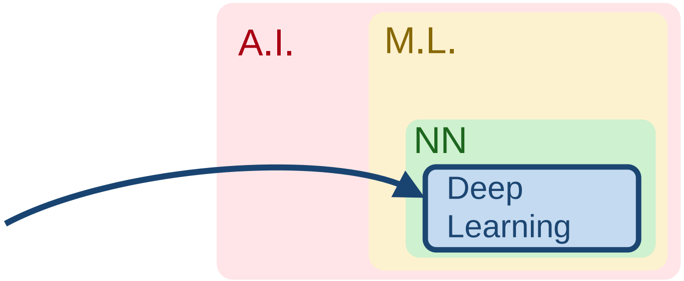


# The Rise of A.I.




- **No.** A real **revolution** is taking shape
- The core driver is not « AI » but **Deep Learning**
- Nearly **all domains of science** have had some subfields which have been **profoundly transformed** by deep learning along the past 10 years

# The Rise of A.I.

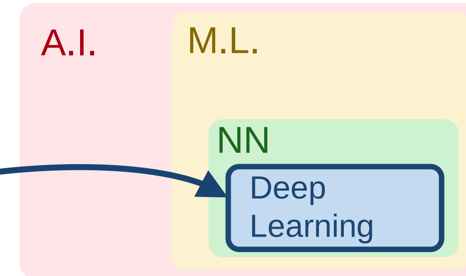



- **No.** A real **revolution** is taking shape
- The core driver is not « AI » but **Deep Learning**
- Nearly **all domains of science** have had some subfields which have been **profoundly transformed** by deep learning along the past 10 years
-

Google Scholar

| | Publication | h5-index | h5-median |
|---|---|---|---|
| 1. | Nature | 444 | 667 |
| 2. | The New England Journal of Medicine | 432 | 780 |
| 3. | Science | 401 | 614 |
| 4. | IEEE/CVF Conference on Computer Vision and Pattern Recognition | 389 | 627 |
| 5. | The Lancet | 354 | 635 |
| 6. | Advanced Materials | 312 | 418 |
| 7. | Nature Communications | 307 | 428 |
| 8. | Cell | 300 | 505 |
| 9. | International Conference on Learning Representations | 286 | 533 |
| 10. | Neural Information Processing Systems | 278 | 436 |

# The Rise of A.I.

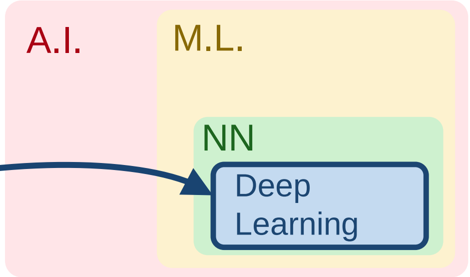



- **No.** A real **revolution** is taking shape
- The core driver is not « AI » but **Deep Learning**
- Nearly **all domains of science** have had some subfields which have been **profoundly transformed** by deep learning along the past 10 years
- Google Scholar

| | Publication | h5-index | h5-median |
|---|---|---|---|
| 1. | Nature | 444 | 667 |
| 2. | The New England Journal of Medicine | 432 | 780 |
| 3. | Science | 401 | 614 |
| 4. | IEEE/CVF Conference on Computer Vision and Pattern Recognition | 389 | 627 |
| 5. | The Lancet | 354 | 635 |
| 6. | Advanced Materials | 312 | 418 |
| 7. | Nature Communications | 307 | 428 |
| 8. | Cell | 300 | 505 |
| 9. | International Conference on Learning Representations | 286 | 533 |
| 10. | Neural Information Processing Systems | 278 | 436 |

M.L. conferences

# The Rise of A.I.



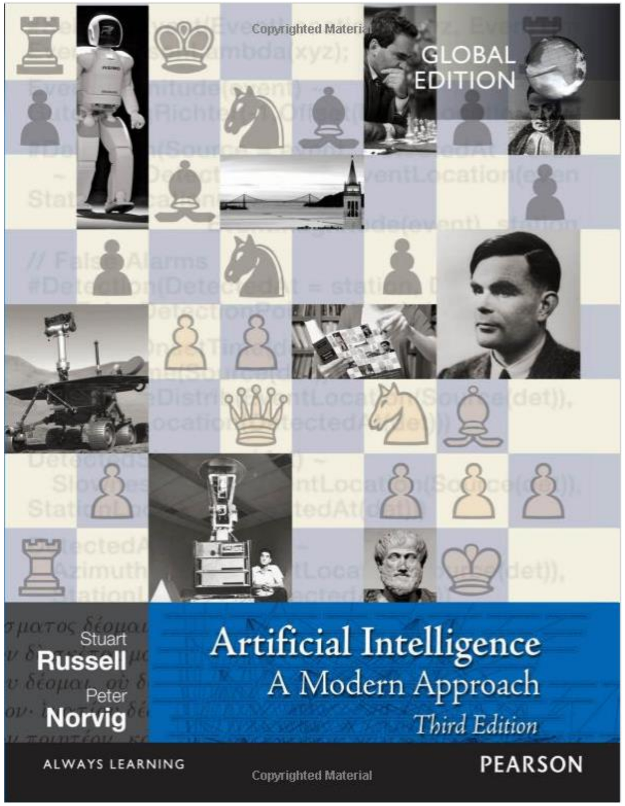

- **No.** A real **revolution** is taking shape
- The core driver is not « AI » but **Deep Learning**
- Nearly **all domains of science** have had some subfields which have been **profoundly transformed** by deep learning along the past 10 years
- Google Scholar

| | Publication | h5-index | h5-median |
|---|---|---|---|
| 1. | Nature | 444 | 667 |
| 2. | The New England Journal of Medicine | 432 | 780 |
| 3. | Science | 401 | 614 |
| 4. | IEEE/CVF Conference on Computer Vision and Pattern Recognition | 389 | 627 |
| 5. | The Lancet | 354 | 635 |
| 6. | Advanced Materials | 312 | 418 |
| 7. | Nature Communications | 307 | 428 |
| 8. | Cell | 300 | 505 |
| 9. | International Conference on Learning Representations | 286 | 533 |
| 10. | Neural Information Processing Systems | 278 | 436 |

M.L. conferences

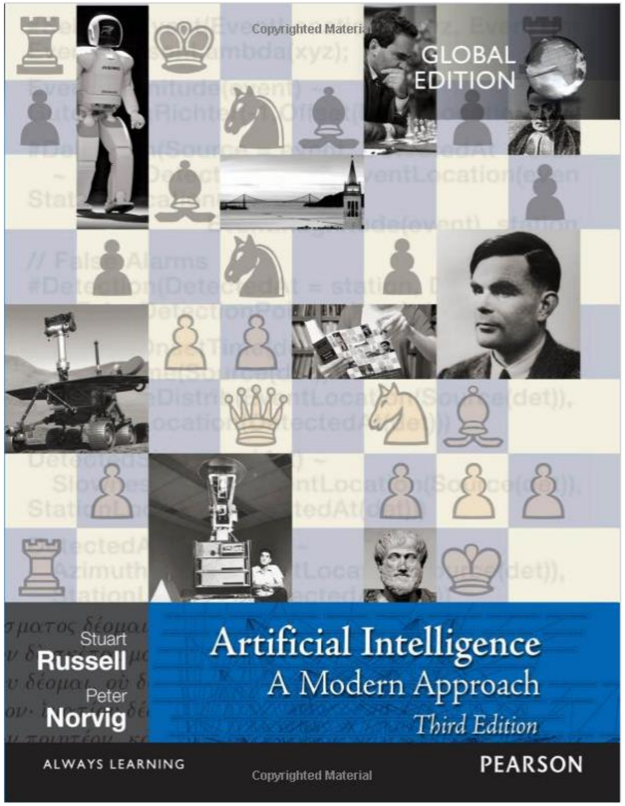
- Likely, nearly all branches of industry, public institutions and professional sectors will soon be profoundly impacted as well

# Applied A.I. *(What A.I. researchers & companies actually do!)*



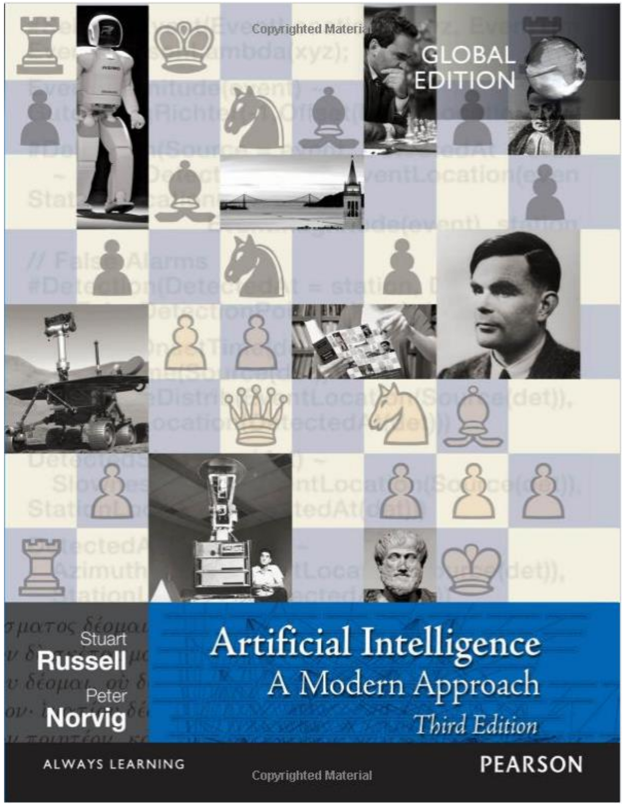
# Applied A.I. *(What A.I. researchers & companies actually do!)*

- Solving Numerical Problems
  - Finding Uncontrained/Constrained Solutions
  - Adversarial Contexts (Game Theory)



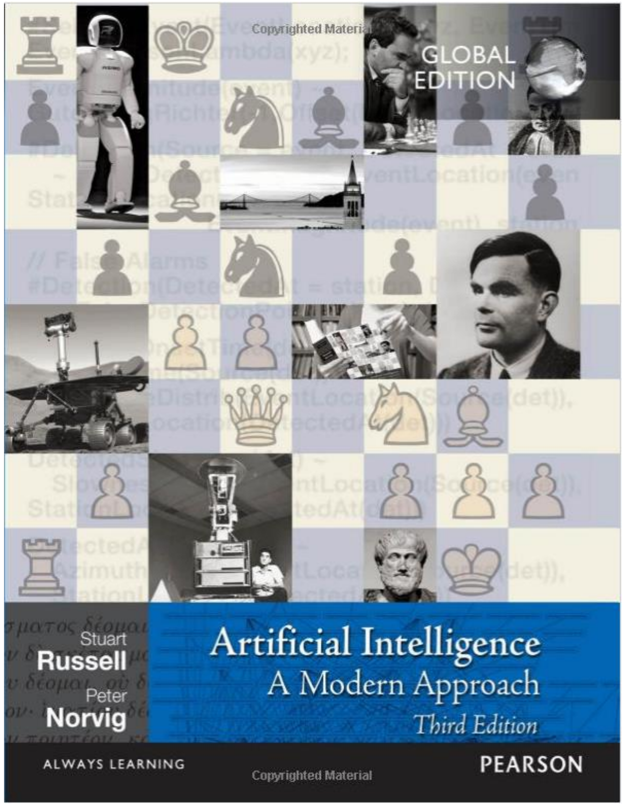
## Applied A.I. *(What A.I. researchers & companies actually do!)*

- Solving Numerical Problems
  - Finding Uncontrained/Constrained Solutions
  - Adversarial Contexts (Game Theory)
- Representing Knowledge and Reasoning
  - Logic, Inference, Planification
  - Ontologies (Symbolic A.I.)



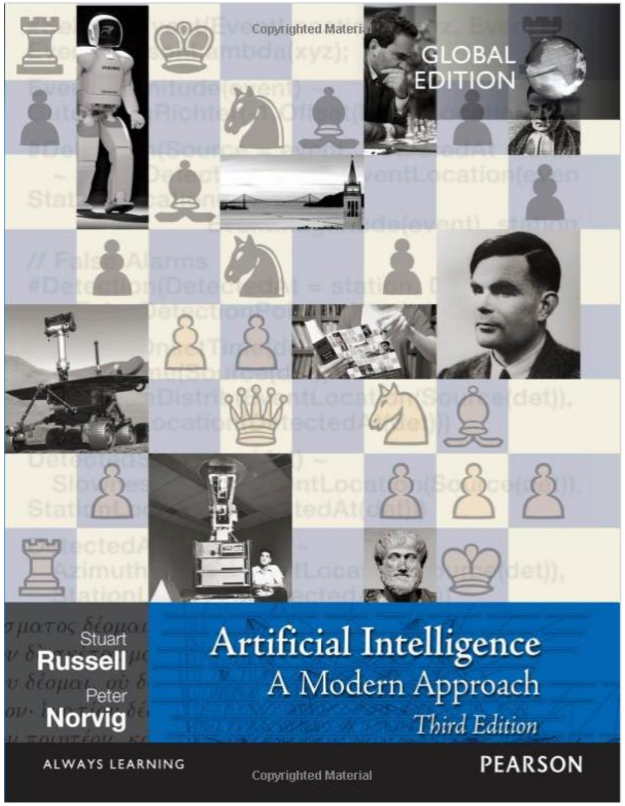
## Applied A.I. *(What A.I. researchers & companies actually do!)*

- Solving Numerical Problems
  - Finding Uncontrained/Constrained Solutions
  - Adversarial Contexts (Game Theory)
- Representing Knowledge and Reasoning
  - Logic, Inference, Planification
  - Ontologies (Symbolic A.I.)
- Managing Uncertainty
  - Representations
  - Probabilistic Models
  - Decision Processes



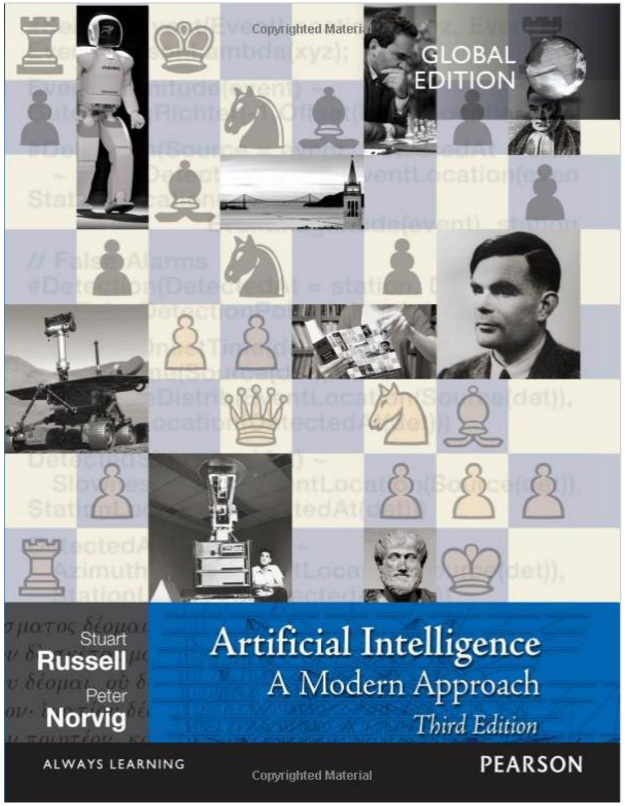
# Applied A.I. *(What A.I. researchers & companies actually do!)*

- Solving Numerical Problems
  - Finding Uncontrained/Constrained Solutions
  - Adversarial Contexts (Game Theory)
- Representing Knowledge and Reasoning
  - Logic, Inference, Planification
  - Ontologies (Symbolic A.I.)
- Managing Uncertainty
  - Representations
  - Probabilistic Models
  - Decision Processes
- Language and Communication
  - Natural Language Processing
  - Speech Recognition, Translation
  - Audio-Visual Synthesis
  - Robotics



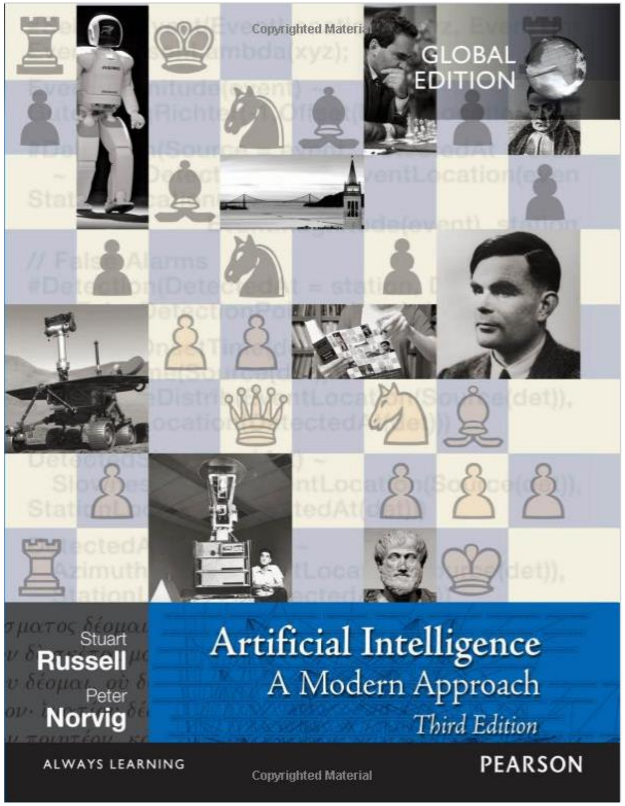
# Applied A.I. *(What A.I. researchers & companies actually do!)*

- Solving Numerical Problems
  - Finding Uncontrained/Constrained Solutions
  - Adversarial Contexts (Game Theory)
- Representing Knowledge and Reasoning
  - Logic, Inference, Planification
  - Ontologies (Symbolic A.I.)
- Managing Uncertainty
  - Representations
  - Probabilistic Models
  - Decision Processes
- Language and Communication
  - Natural Language Processing
  - Speech Recognition, Translation
  - Audio-Visual Synthesis
  - Robotics
- Machine Learning



# Applied A.I. *(What A.I. researchers & companies actually do!)*

- Solving Numerical Problems
  - Finding Uncontrained/Constrained Solutions
  - Adversarial Contexts (Game Theory)
- Representing Knowledge and Reasoning
  - Logic, Inference, Planification
  - Ontologies (Symbolic A.I.)
- Managing Uncertainty
  - Representations
  - Probabilistic Models
  - Decision Processes
- Language and Communication
  - Natural Language Processing
  - Speech Recognition, Translation
  - Audio-Visual Synthesis
  - Robotics
- Machine Learning



## A.I. Philosophy

- A fascinating field

# A.I. Philosophy

- A fascinating field
- Relatively **niche** compared to *applied A.I.*

# A.I. Philosophy

- A fascinating field
- Relatively **niche** compared to *applied A.I.*
- At the intersection of Philosophy, Futurology, Social Sciences, Psychology, Logic and (sometimes) Computer Science

# A.I. Philosophy

- A fascinating field
- Relatively **niche** compared to *applied A.I.*
- At the intersection of Philosophy, Futurology, Social Sciences, Psychology, Logic and (sometimes) Computer Science
- Some subtopics:
  - AI Safety / AI Risk / AI Alignment
  - AI Ethics / AI Bias / AI Fairness
  - Conciousness / Sentience / Free Will
  - Definitions of Intelligence

# A.I. Philosophy

Different levels of A.I. are distinguished:

- **Specialized** A.I.
- Artificial **General** Intelligence (AGI)
- **Human-Level** Artificial Intelligence
- Artificial **Super-Intelligence** (ASI)

# A.I. Philosophy

Different levels of A.I. are distinguished:

- **Specialized** A.I.
- Artificial **General** Intelligence (AGI)
- **Human-Level** Artificial Intelligence
- Artificial **Super-Intelligence** (ASI)

- Very clearly where we are **now**
- Can be cast into actual engineering questions and products

# A.I. Philosophy

Different levels of A.I. are distinguished:

- **Specialized** A.I.
- Artificial **General** Intelligence (AGI)
- **Human-Level** Artificial Intelligence
- Artificial **Super-Intelligence** (ASI)

- Very clearly where we are **now**
- Can be cast into actual engineering questions and products

Does **not** exist (yet) !
No agreement on:

- Is it **achievable ?**
- **How** to achieve it ?
- **When** will we achieve it?
- **Should we** achieve it?

# A.I. Philosophy

Different levels of A.I. are distinguished:

- **Specialized** A.I.
- Artificial **General** Intelligence (AGI)
- **Human-Level** Artificial Intelligence
- Artificial **Super-Intelligence** (ASI)

Most A.I. researchers agree that to safely deploy such systems, we need to **align them** (*A.I. alignment research*)

- Very clearly where we are **now**
- Can be cast into actual engineering questions and products

Does **not** exist (yet) !
No agreement on:

- Is it **achievable ?**
- **How** to achieve it ?
- **When** will we achieve it?
- **Should we** achieve it?

# A.I. Philosophy

Different levels of A.I. are distinguished:

- **Specialized** A.I.
- Artificial **General** Intelligence (AGI)
- **Human-Level** Artificial Intelligence
- Artificial **Super-Intelligence** (ASI)

- Very clearly where we are **now**
- Can be cast into actual engineering questions and products

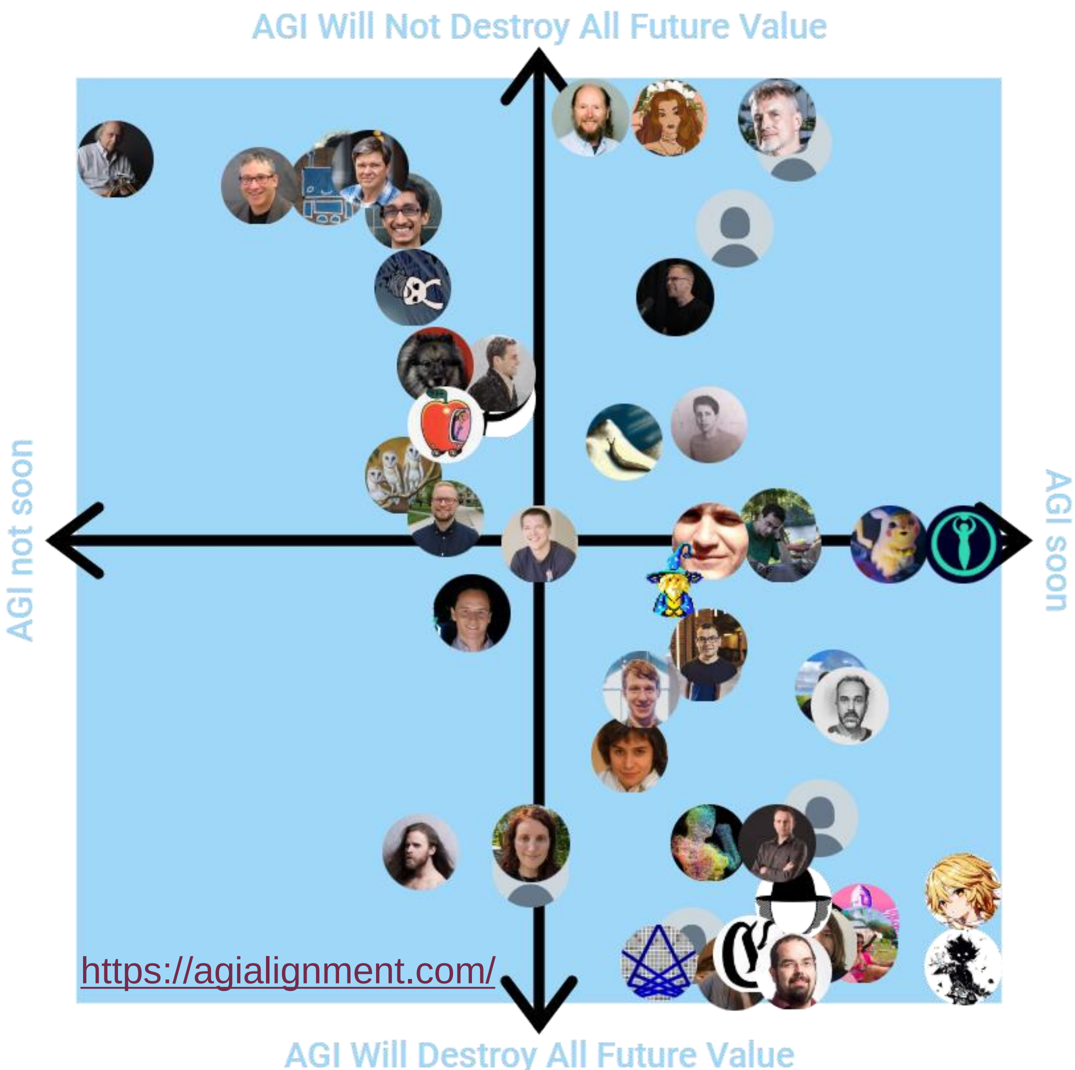
Does **not** exist (yet) !
No agreement on:

- Is it **achievable ?**
- **How** to achieve it ?
- **When** will we achieve it?
- **Should we** achieve it?

Most A.I. researchers agree that to safely deploy such systems, we need to **align them** (*A.I. alignment research*)

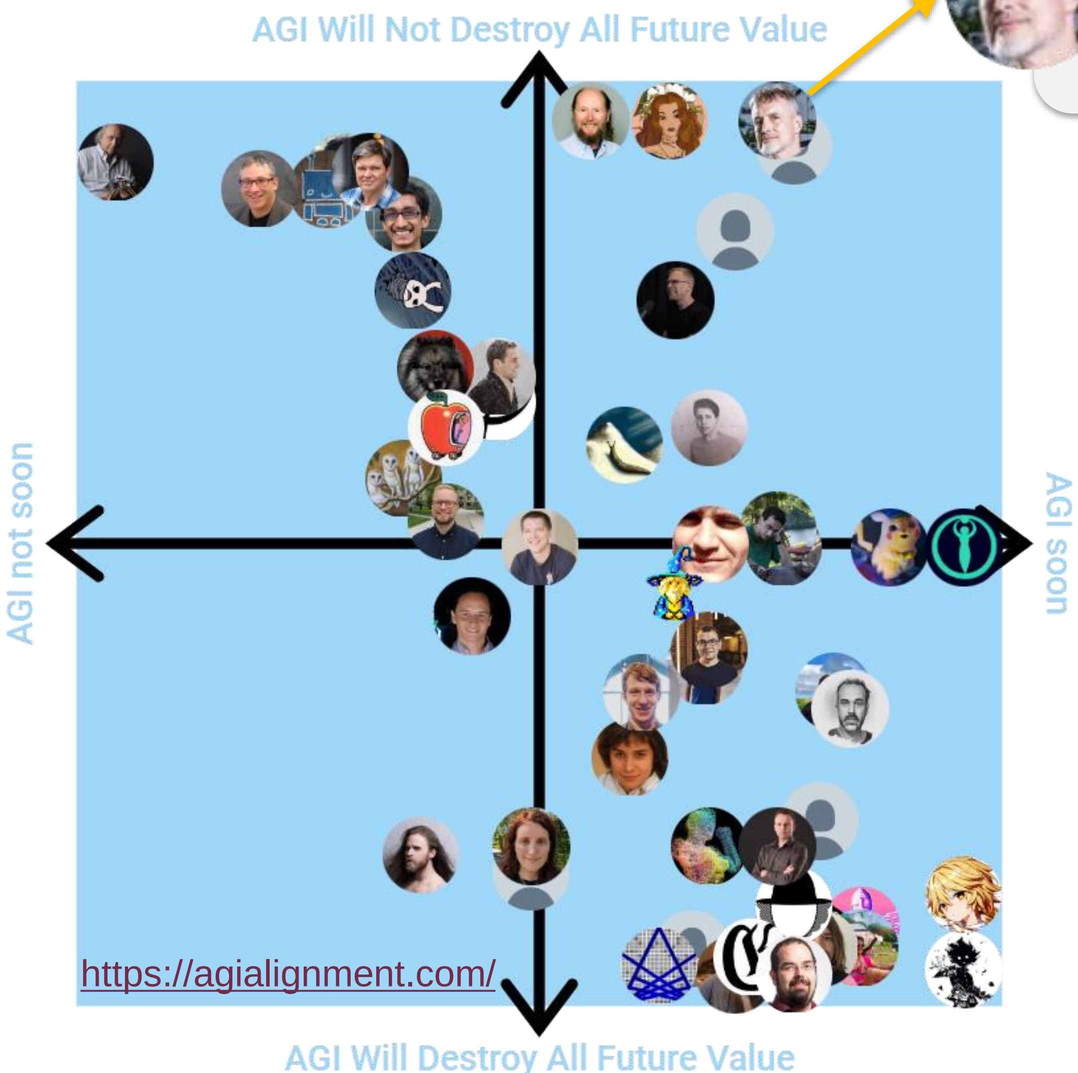
*They disagree on **how hard** it is, and on **how much time** we have to figure it out!*

# A.I. Philosophy



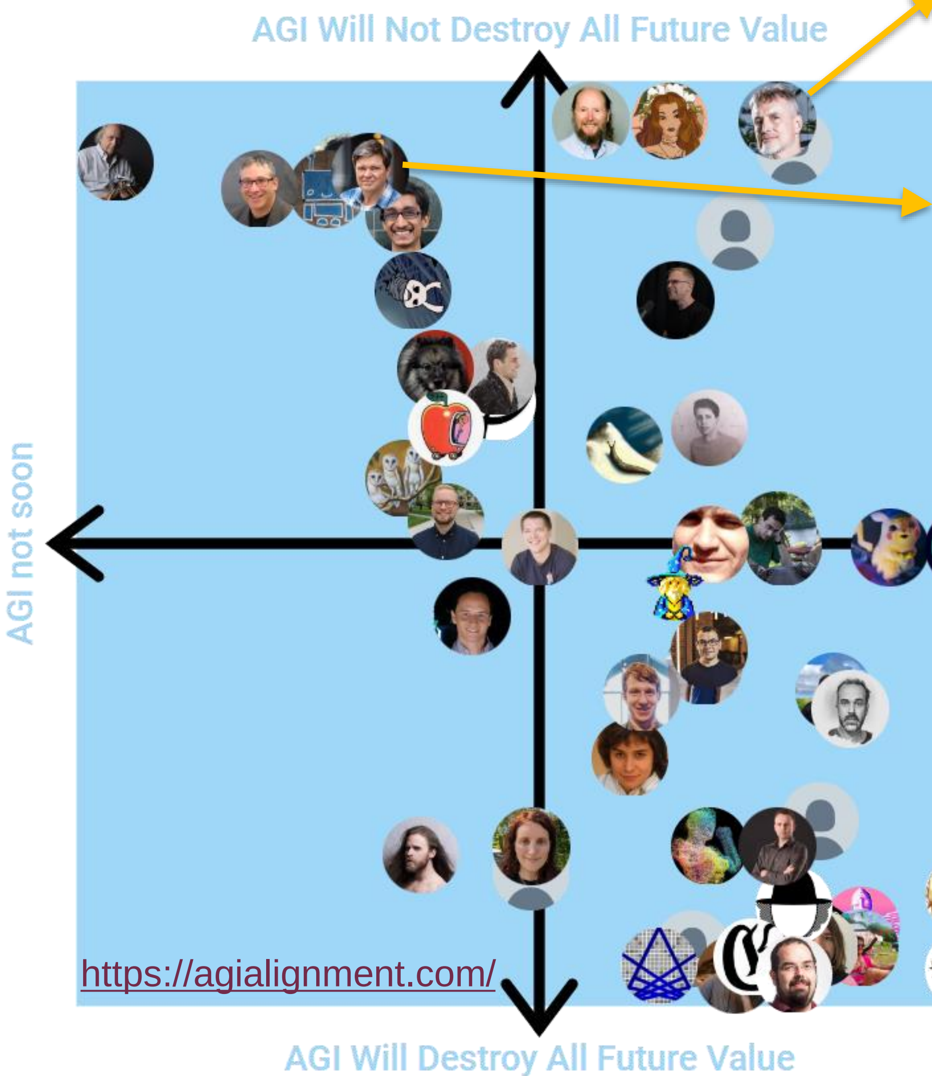

https://agialignment.com/

# A.I. Philosophy




- Jürgen Schmidhuber, DM Inst. for AI Research (Switz.), LSTM inventor
- *Annotated History of Modern AI and Deep Learning*

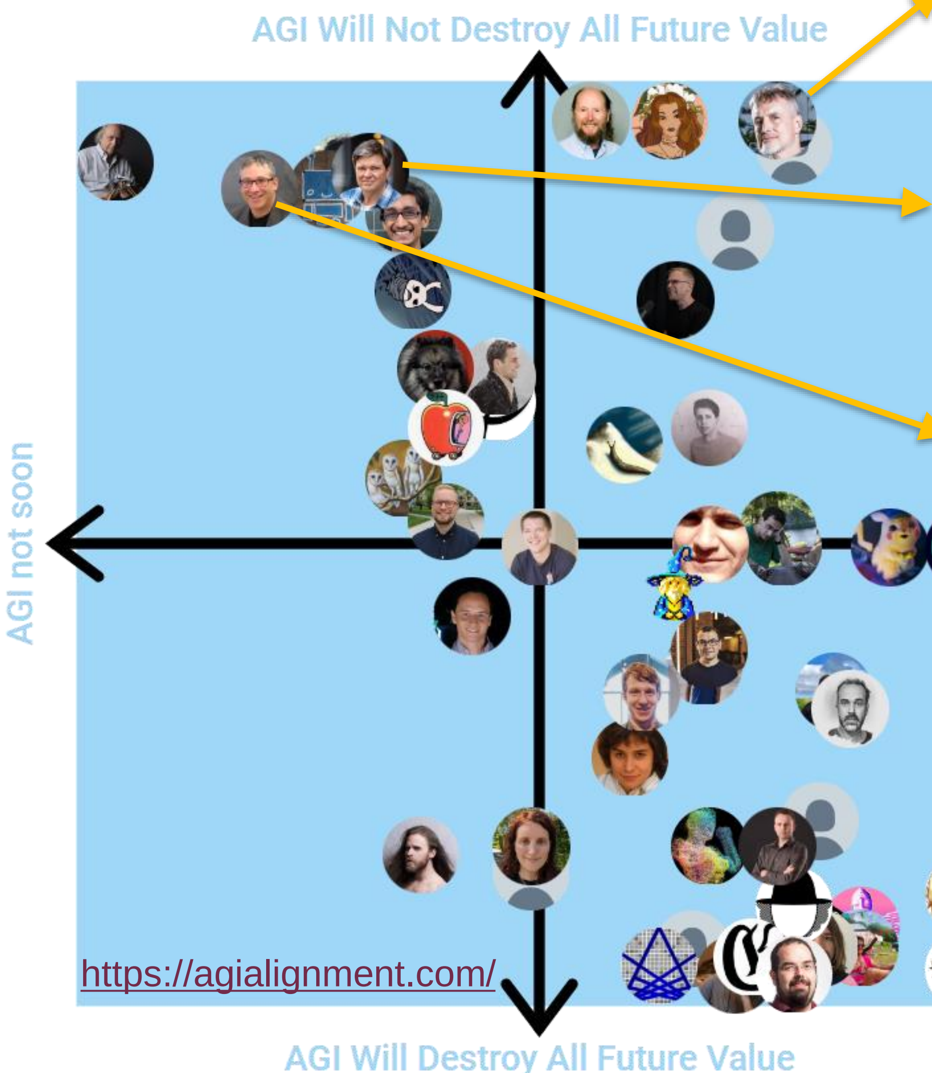
# A.I. Philosophy




- Jürgen Schmidhuber, DM Inst. for AI Research (Switz.), LSTM inventor
- *Annotated History of Modern AI and Deep Learning*

- Yann Lecun, Chief AI Scientist at Meta AI, NYU Prof.
- *A Path Towards Autonomous Machine Intelligence*

# A.I. Philosophy

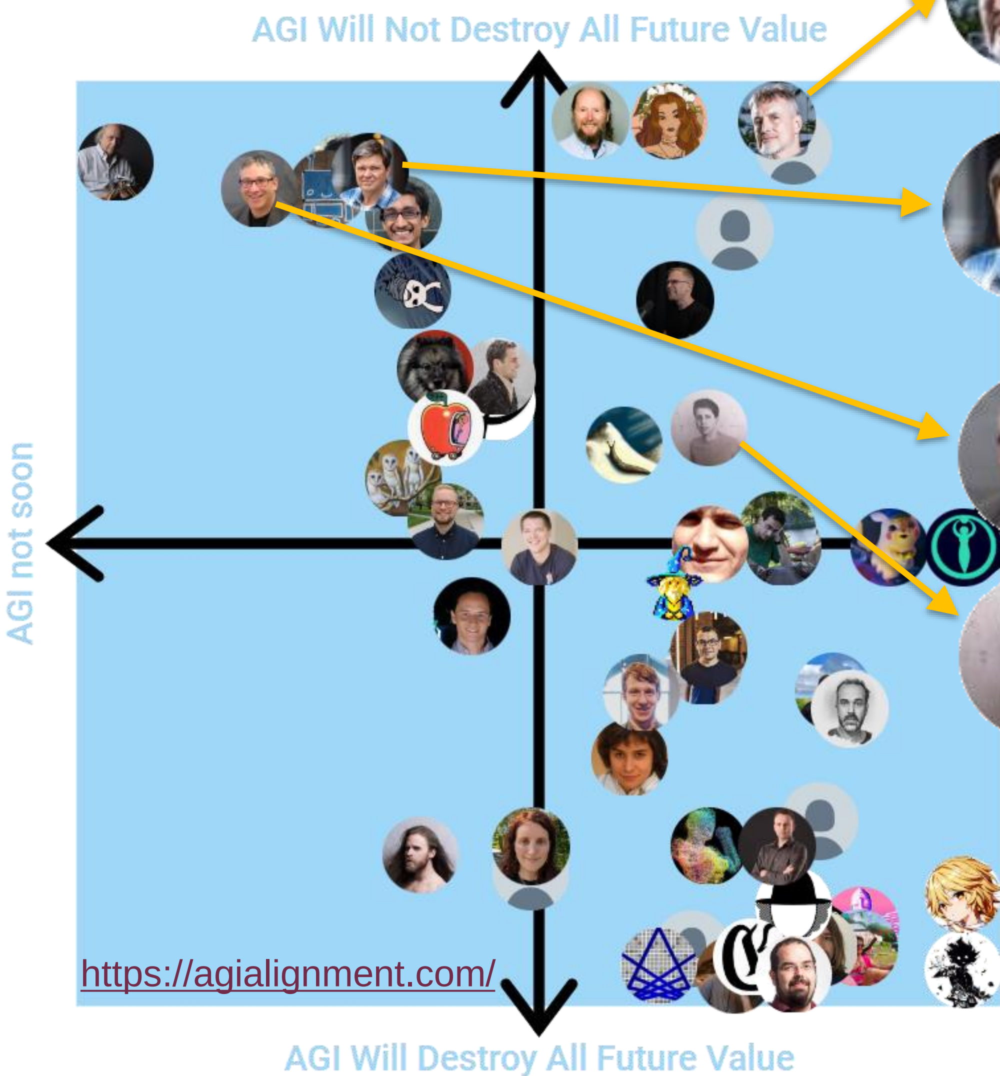



- Jürgen Schmidhuber, DM Inst. for AI Research (Switz.), LSTM inventor
- *Annotated History of Modern AI and Deep Learning*

- Yann Lecun, Chief AI Scientist at Meta AI, NYU Prof.
- *A Path Towards Autonomous Machine Intelligence*

- Gary Marcus, NYU Prof. Emeritus, book author
- *Blog: The Road to AI We Can Trust*

# A.I. Philosophy



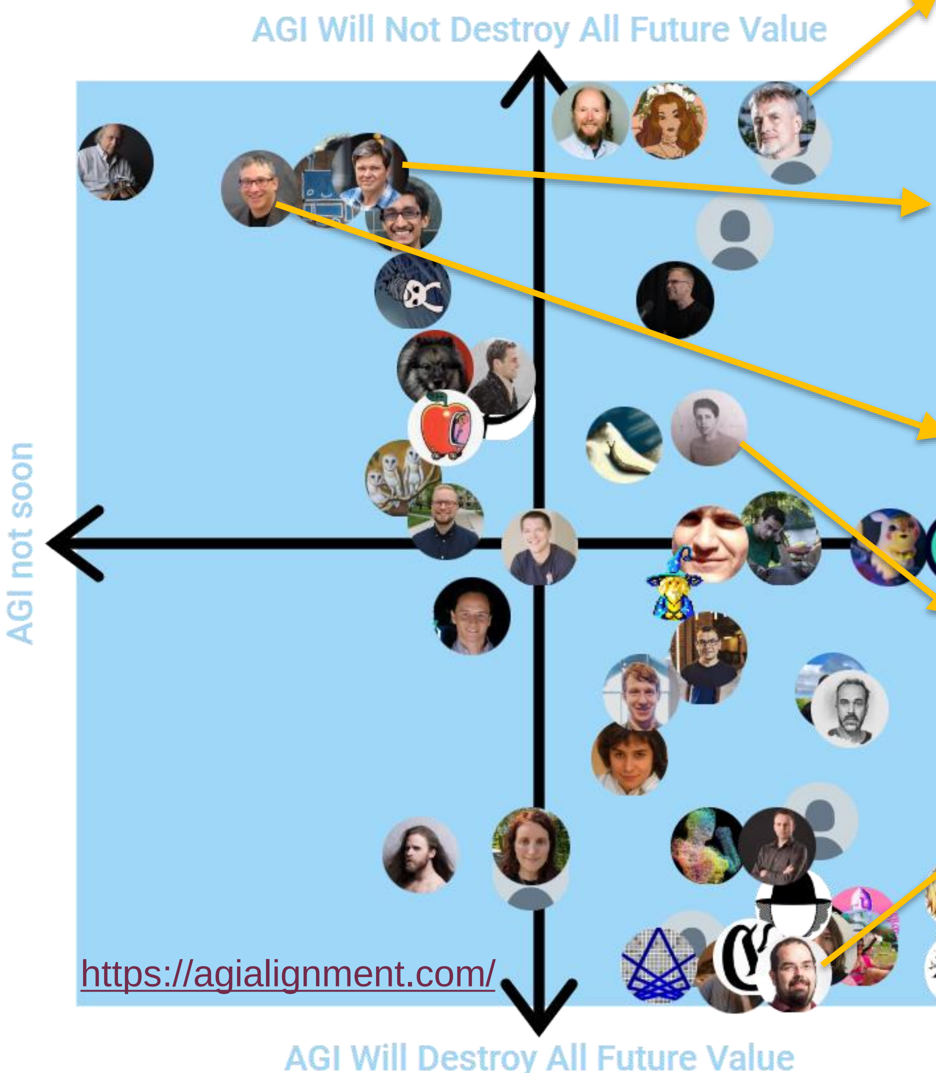

- Jürgen Schmidhuber, DM Inst. for AI Research (Switz.), LSTM inventor
- *Annotated History of Modern AI and Deep Learning*

- Yann Lecun, Chief AI Scientist at Meta AI, NYU Prof.
- *A Path Towards Autonomous Machine Intelligence*

- Gary Marcus, NYU Prof. Emeritus, book author
- *Blog: The Road to AI We Can Trust*

- Sam Altman, CEO and co-founder at OpenAI
- *Planning for AGI and beyond*

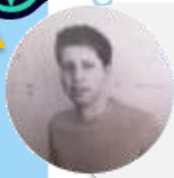
# A.I. Philosophy




- Jürgen Schmidhuber, DM Inst. for AI Research (Switz.), LSTM inventor
- *Annotated History of Modern AI and Deep Learning*

- Yann Lecun, Chief AI Scientist at Meta AI, NYU Prof.
- *A Path Towards Autonomous Machine Intelligence*

- Gary Marcus, NYU Prof. Emeritus, book author
- *Blog: The Road to AI We Can Trust*



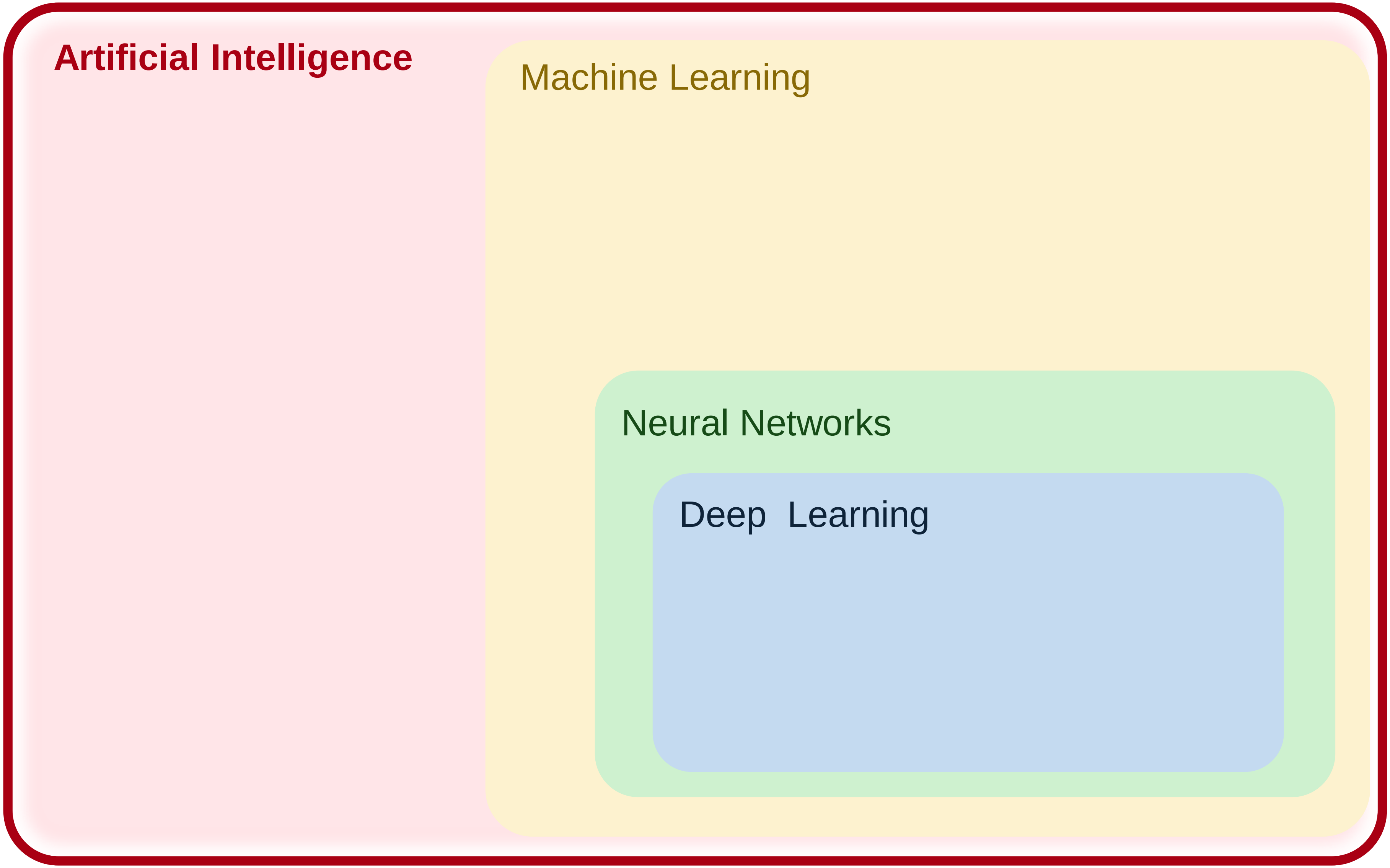
- Sam Altman, CEO and co-founder at OpenAI
- *Planning for AGI and beyond*

- Eliezer Yudkowski, researcher and co-founder at the MIRI, author of more than 300 blogpost + books
- Leading figure in AI alignment
- AGI Ruin: A List of Lethalities

# A.I. Philosophy

Other worthwhile reads on A.I. Philosophy

- *François Chollet's The Implausibility of Intelligence Explosion (2017)*
- *David Chalmer's Could a large Language Model be Concious? (2022)*
- *Scott Aaronson's blog*, a theoretical quantum computer scientist at the University of Texas Austin who took a sabbatical year to work on AI alignment at OpenAI.
- *Nick Bolstrom's "Superintelligence, Paths, Dangers, Strategies" (2014)*


Artificial Intelligence
Machine Learning
Neural Networks
Deep Learning

**Artificial Intelligence**

- Strongly embedded in collective imagination

Machine Learning

Neural Networks

Deep Learning

## Artificial Intelligence

- Strongly embedded in collective imagination
- A *catch-all* term

### Machine Learning

#### Neural Networks

##### Deep Learning

**Artificial Intelligence**

- Strongly embedded in collective imagination
- A *catch-all* term
- Used more in marketing than in actual research & engineering

Machine Learning

Neural Networks

Deep Learning

**Artificial Intelligence**

- Strongly embedded in collective imagination
- A *catch-all* term
- Used more in marketing than in actual research & engineering
- A subfield of *Applied A.I.* is currently sparking a revolution, in science & beyond

Machine Learning

Neural Networks

Deep Learning

**Artificial Intelligence**

- Strongly embedded in collective imagination
- A *catch-all* term
- Used more in marketing than in actual research & engineering
- A subfield of *Applied A.I.* is currently sparking a revolution, in science & beyond
- *A.I. philosophy* is fascinating (and we should probably care)

Machine Learning

Neural Networks

Deep Learning

## Artificial Intelligence

- Strongly embedded in collective imagination
- A *catch-all* term
- Used more in marketing than in actual research & engineering
- A subfield of *Applied A.I.* is currently sparking a revolution, in science & beyond
- *A.I. philosophy* is fascinating (and we should probably care)

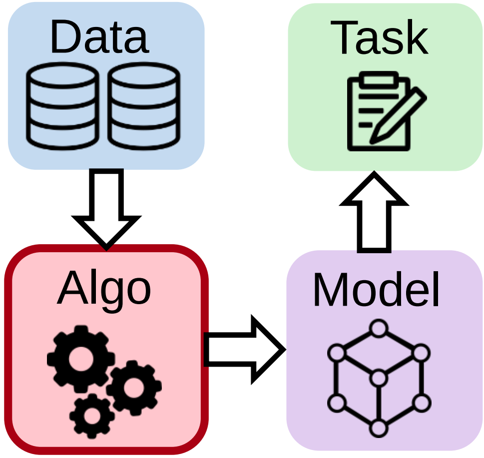
**Machine Learning**

Neural Networks

Deep Learning

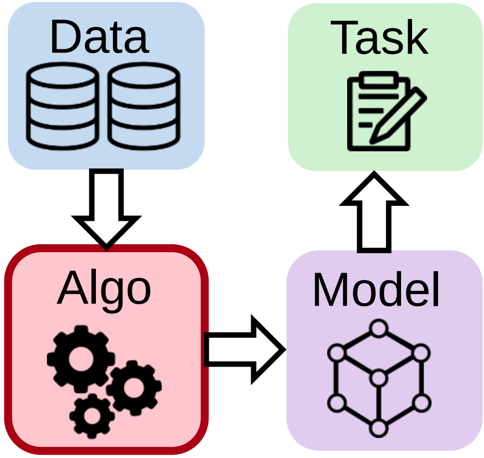
## Definition

The study of **Algorithms** that build **Models** from **Data** to achieve **Tasks**.



# Definition

> The study of **Algorithms** that build **Models** from **Data** to achieve **Tasks**.

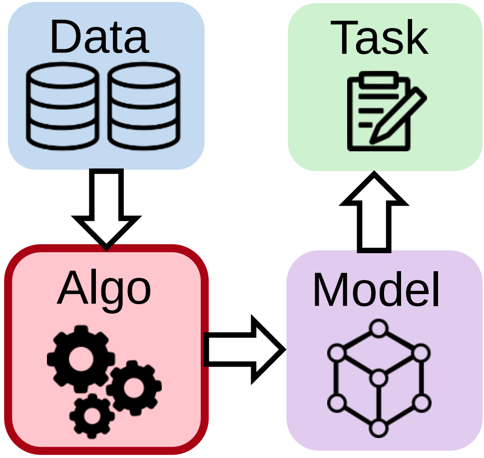



**Tasks:**

- Sort / Visualize / Represent the **Data**

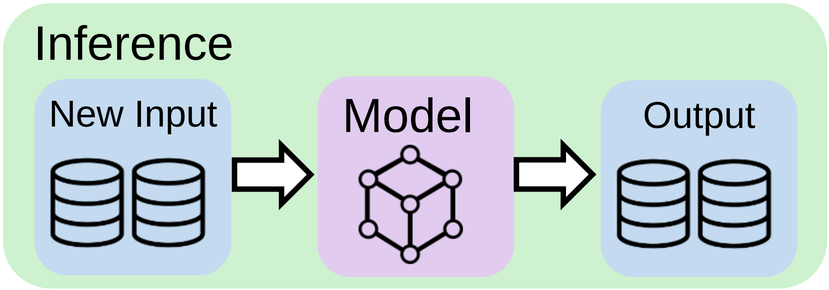
# Definition

The study of **Algorithms** that build **Models** from **Data** to achieve **Tasks**.



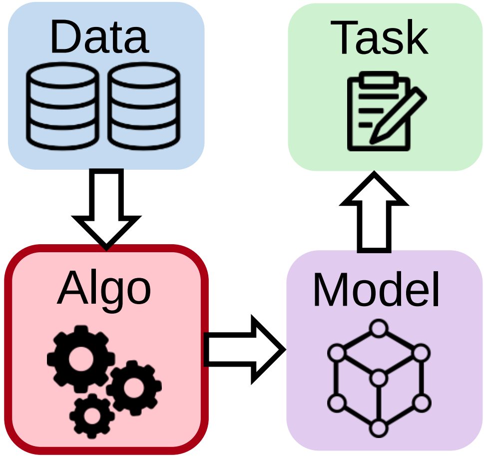

**Tasks:**

- Sort / Visualize / Represent the **Data**
- **Infer** from **new input Data:**
  - Estimate explanatory quantities
  - Generate, complete, predict
  - Transform, convert, translate
  - Decide

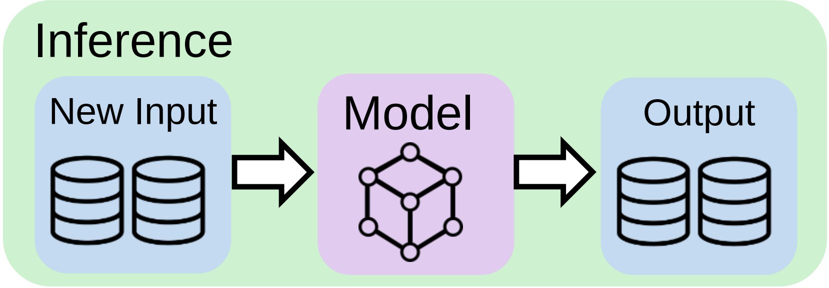


# Definition

The study of **Algorithms** that build **Models** from **Data** to achieve **Tasks**.

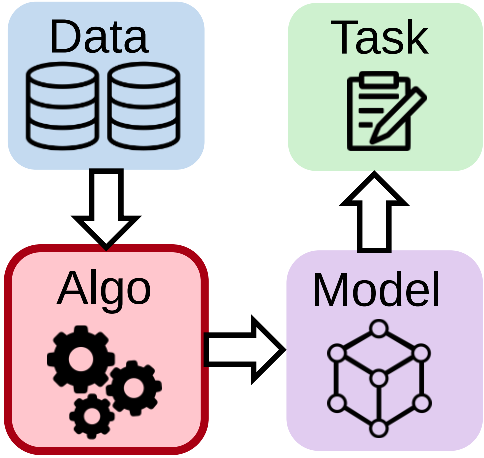



**Tasks:**
- Sort / Visualize / Represent the **Data**
- **Infer** from **new input Data:**
  - Estimate explanatory quantities
  - Generate, complete, predict
  - Transform, convert, translate
  - Decide




**Model** (in machine learning)**:**
- A *“program created by a program”*
- Can be **deterministic** or **stochastic**
- **Data-Driven** as opposed to **Physics- / Knowledge-Driven** model

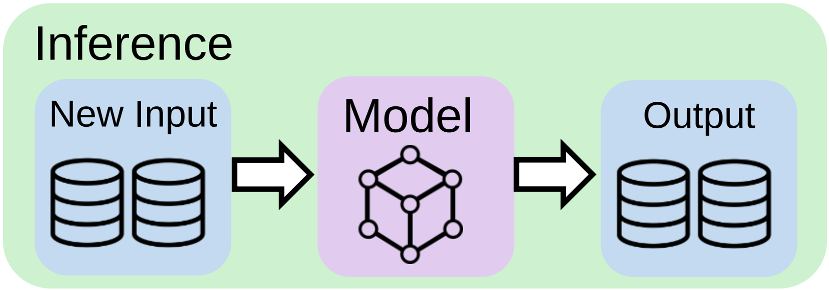
# Definition




The study of **Algorithms** that build **Models** from **Data** to achieve **Tasks**.

**Tasks:**

- Sort / Visualize / Represent the **Data**
- **Infer** from **new input Data:**
  - Estimate explanatory quantities
  - Generate, complete, predict
  - Transform, convert, translate
  - Decide



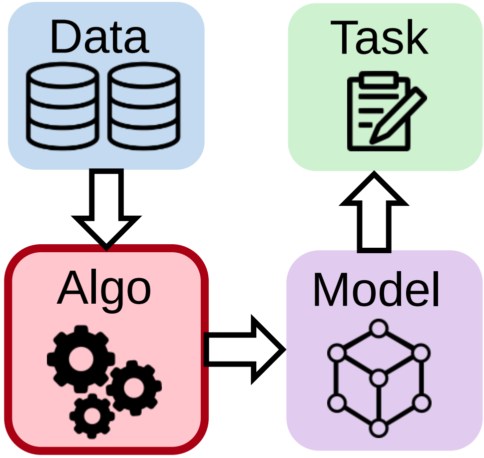

**Model** (in machine learning)**:**

- A *"program created by a program"*
- Can be **deterministic** or **stochastic**
- **Data-Driven** as opposed to **Physics- / Knowledge-Driven** model

Not considered ML, even if the same Task is performed!

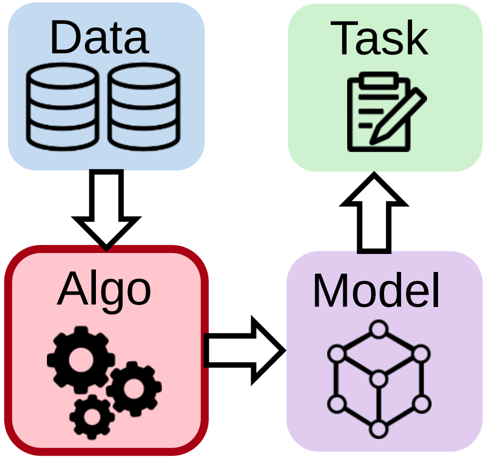
# Main characteristics

- We do not require **explicit concepts** with **meanings** inside the model



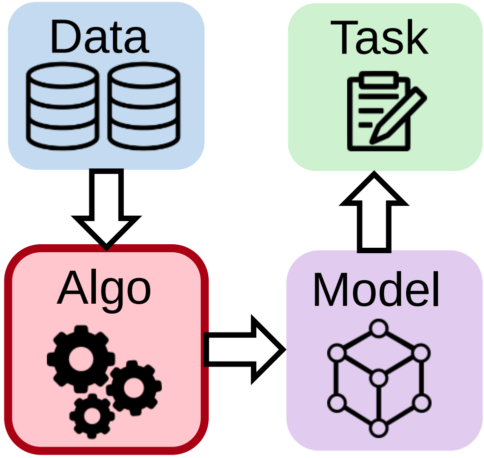
# Main characteristics

- We do not require **explicit concepts** with **meanings** inside the model
  - “Black Box” effect



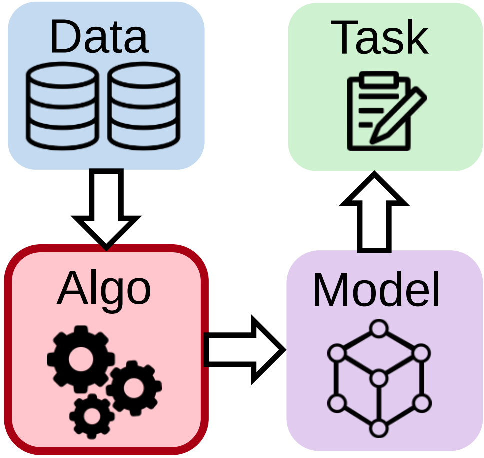
# Main characteristics

- We do not require **explicit concepts** with **meanings** inside the model
  - “Black Box” effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models



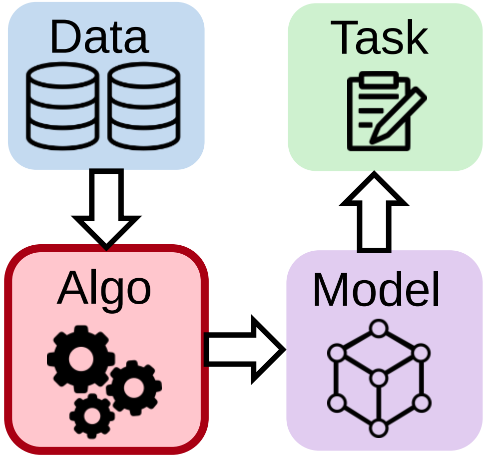
# Main characteristics

- We do not require **explicit concepts** with **meanings** inside the model
  - “Black Box” effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*

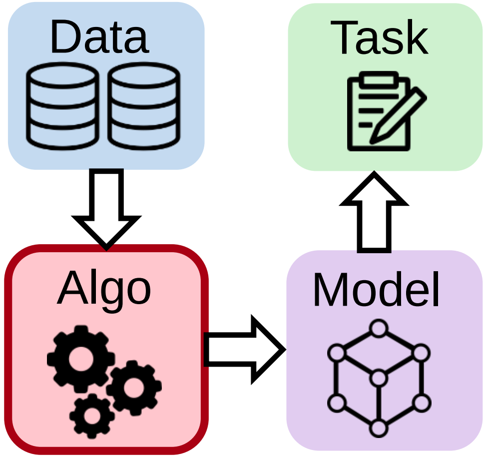


# Main characteristics

- We do not require **explicit concepts** with **meanings** inside the model
  - "Black Box" effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.

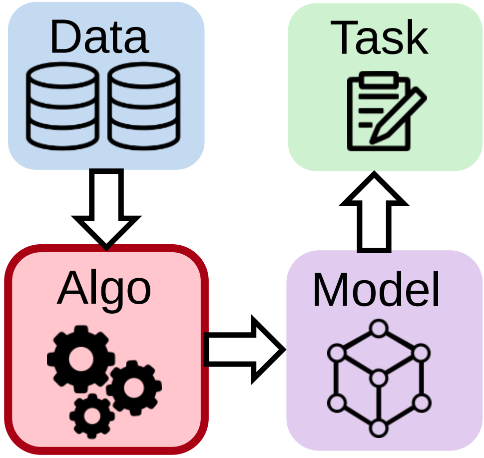


# Main characteristics



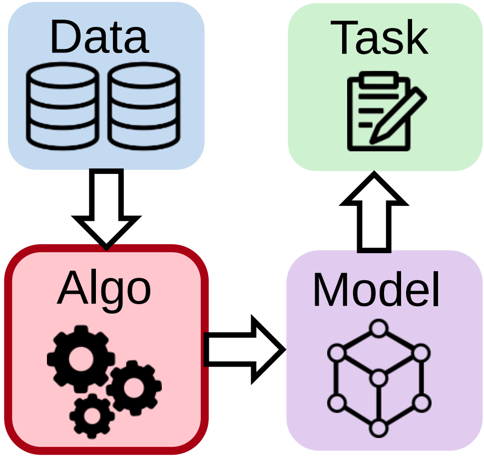

- We do not require **explicit concepts** with **meanings** inside the model
  - “Black Box” effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha

# Main characteristics



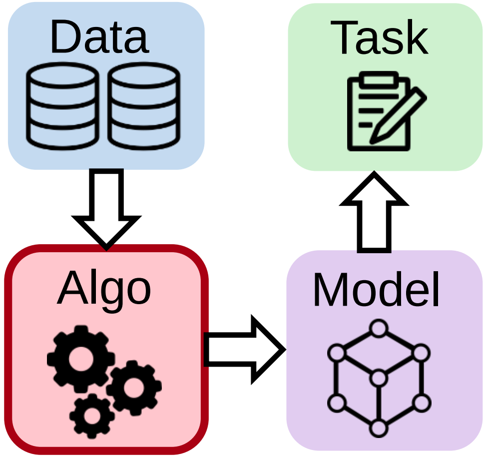

- We do not require **explicit concepts** with **meanings** inside the model
  - "Black Box" effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha
  - Logic-Based approaches, Fuzzy logic

## Main characteristics



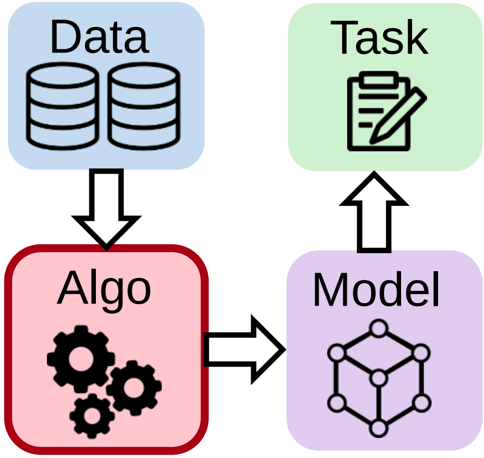

- We do not require **explicit concepts** with **meanings** inside the model
  - "Black Box" effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha
  - Logic-Based approaches, Fuzzy logic
  - Chess engine Deep Blue (explicit rules)

## Main characteristics



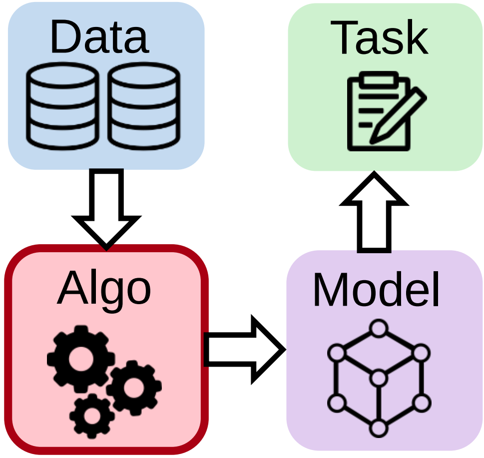

- We do not require **explicit concepts** with **meanings** inside the model
  - "Black Box" effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha
  - Logic-Based approaches, Fuzzy logic
  - Chess engine Deep Blue (explicit rules)
- Is Machine Learning **enough** for A.I.?

# Main characteristics




- We do not require **explicit concepts** with **meanings** inside the model
  - “Black Box” effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha
  - Logic-Based approaches, Fuzzy logic
  - Chess engine Deep Blue (explicit rules)
- Is Machine Learning **enough** for A.I.?
  - A heated debate! (ex: Yann LeCun vs. Gary Marcus)

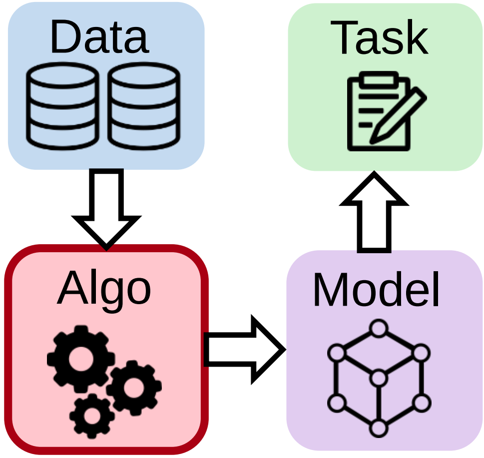
# Main characteristics




- We do not require **explicit concepts** with **meanings** inside the model
  - “Black Box” effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha
  - Logic-Based approaches, Fuzzy logic
  - Chess engine Deep Blue (explicit rules)

- Is Machine Learning **enough** for A.I.?
  - A heated debate! (ex: Yann LeCun vs. Gary Marcus)
  - *Neuro-symbolic A.I.* aims at combining Symbolic A.I. and Neural Networks

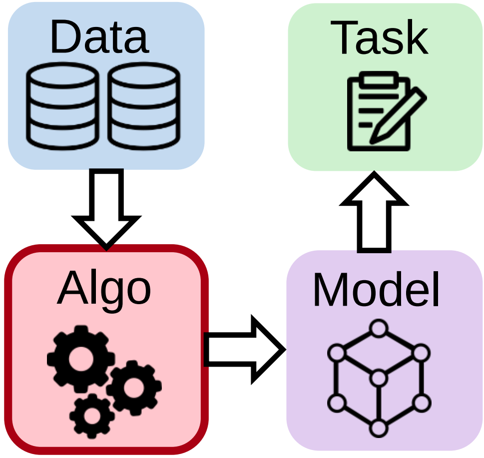
# Main characteristics




- We do not require **explicit concepts** with **meanings** inside the model
  - "Black Box" effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha
  - Logic-Based approaches, Fuzzy logic
  - Chess engine Deep Blue (explicit rules)

- Is Machine Learning **enough** for A.I.?
  - A heated debate! (ex: Yann LeCun vs. Gary Marcus)
  - *Neuro-symbolic A.I.* aims at combining Symbolic A.I. and Neural Networks
  - This course will focus on **machine learning**

## Main characteristics



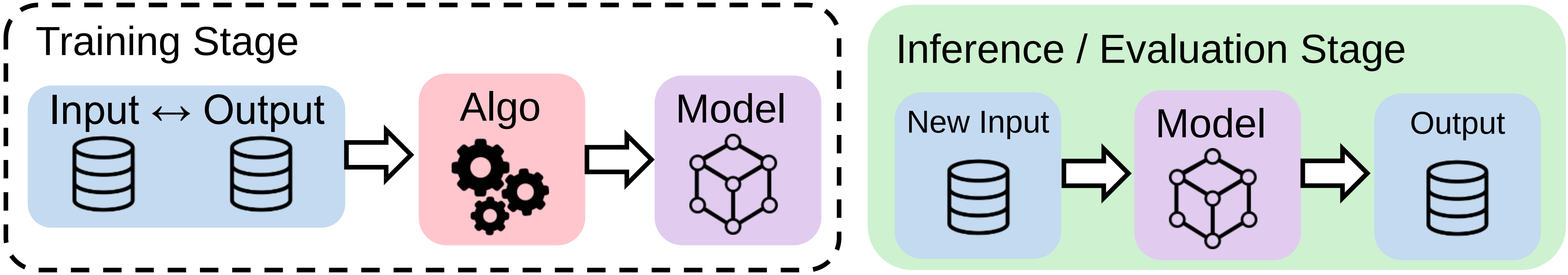

- We do not require **explicit concepts** with **meanings** inside the model
  - "Black Box" effect
  - A rich subfield of ML dedicated to explainability / interpretability / theoretical guarantees on models
  - But this is done *a posteriori*
- Opposed to the ***Symbolic*** Vision of A.I.
  - Ex: Programming languages of higher and higher level
    Assembly → C → Java → Python → Mathematica / Wolfram Alpha
  - Logic-Based approaches, Fuzzy logic
  - Chess engine Deep Blue (explicit rules)

- Is Machine Learning **enough** for A.I.?
  - A heated debate! (ex: Yann LeCun vs. Gary Marcus)
  - *Neuro-symbolic A.I.* aims at combining Symbolic A.I. and Neural Networks
  - This course will focus on **machine learning**
  - Promising approach: Hybridizing **physics-** and **data-driven** models

# Types of ML based on available data

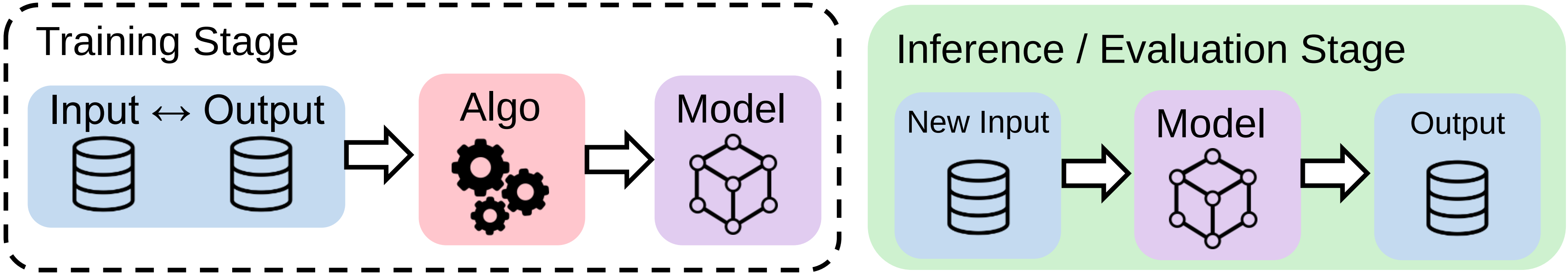
# Types of ML based on available data

- Supervised Learning

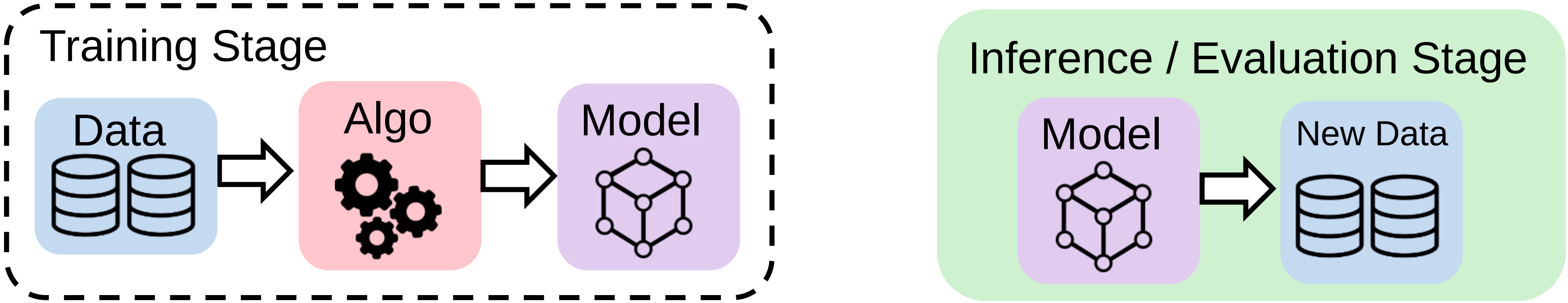


# Types of ML based on available data

- Supervised Learning

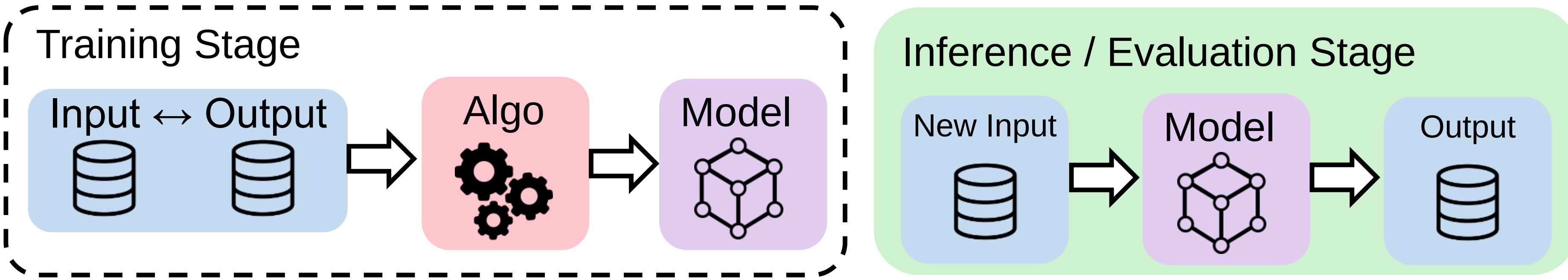



- Unsupervised Learning

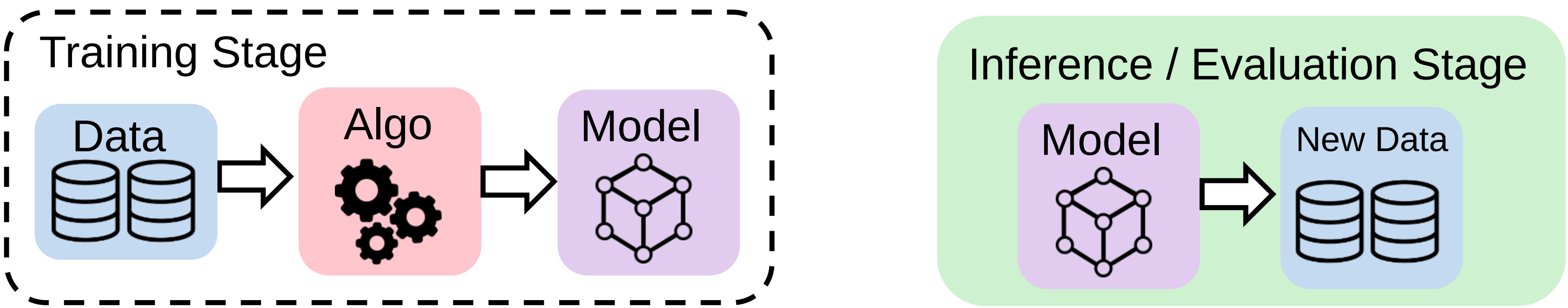


# Types of ML based on available data

- Supervised Learning




- Unsupervised Learning

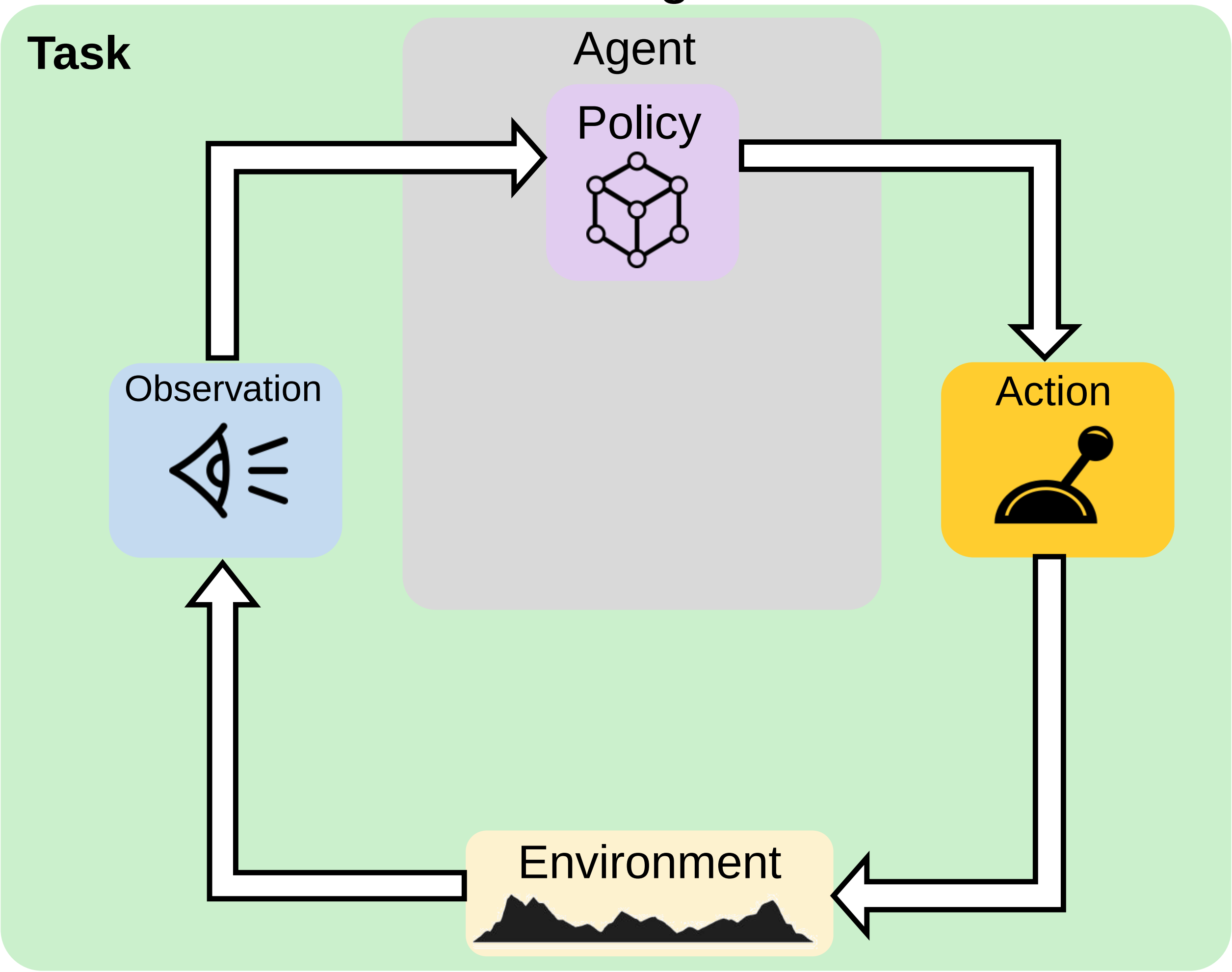



- *Gray Area:*

  Semi-supervised learning, self-supervised learning, weak labels,…

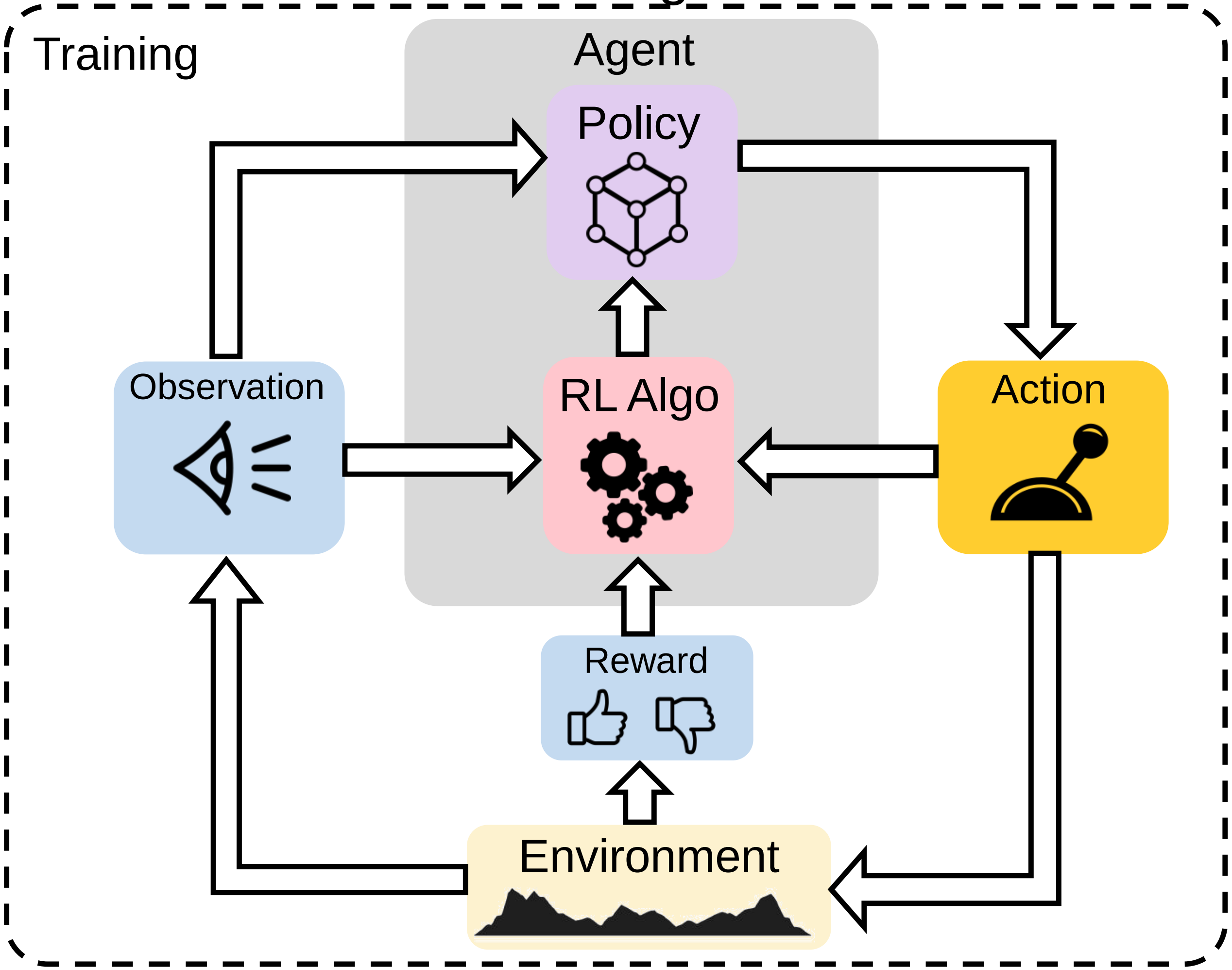
# Types of ML based on available data

- Reinforcement Learning



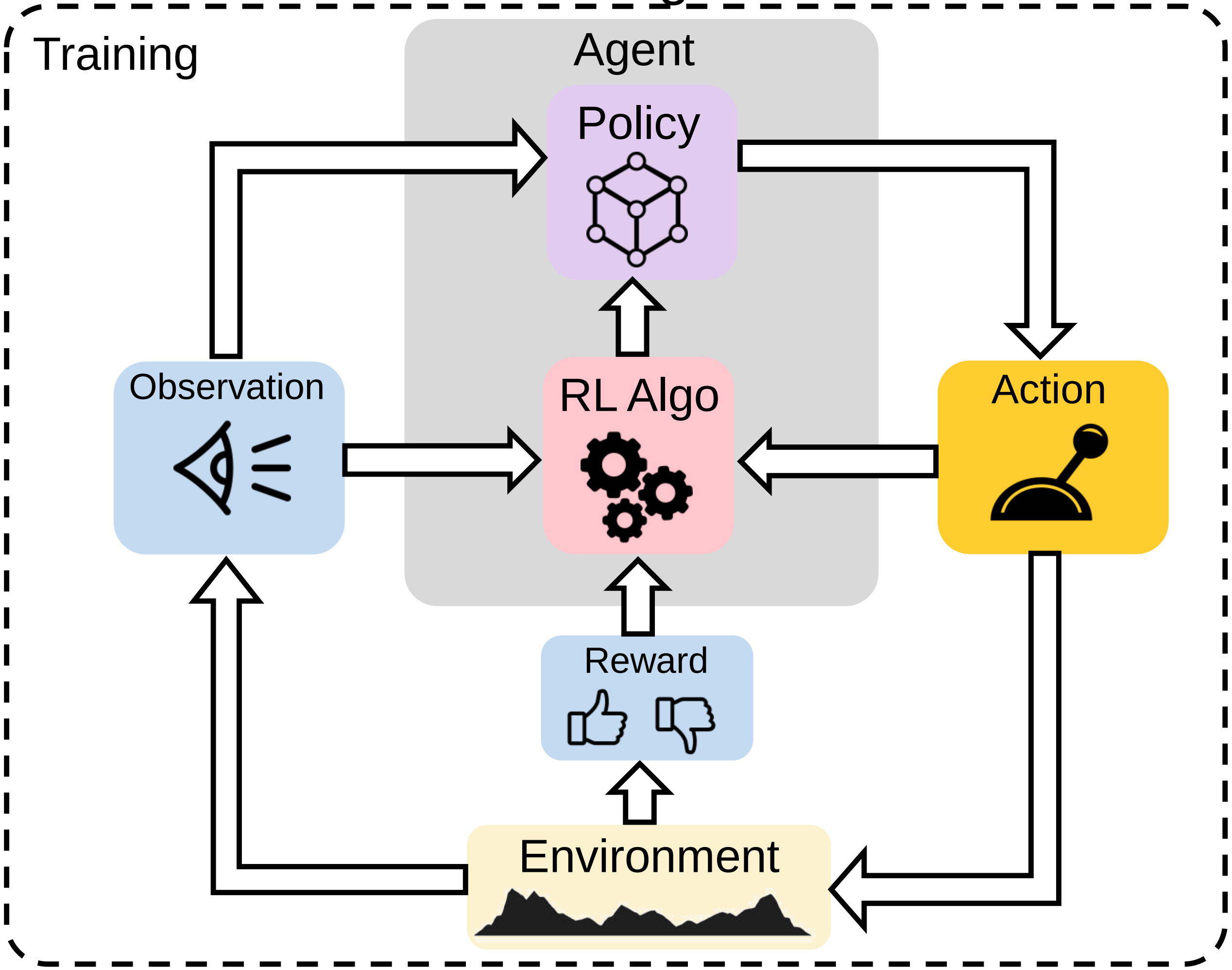
# Types of ML based on available data

- Reinforcement Learning



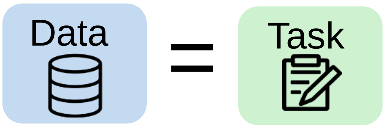
# Types of ML based on available data

- Reinforcement Learning




## Others

- **Active learning:** Choose on which samples to learn
- **Meta Learning:** Learn how to learn, on a set of **tasks**




- **Continual Learning:** Also called *lifelong* learning

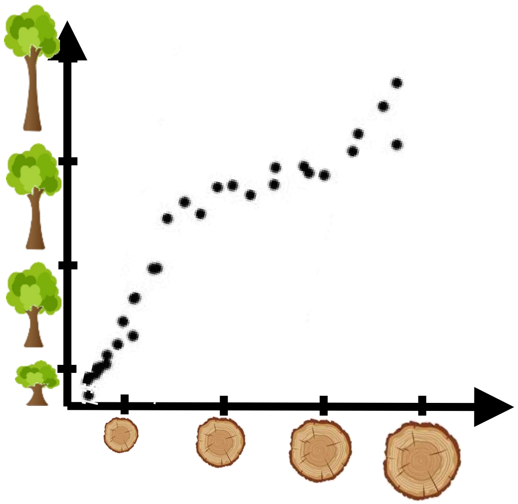

# Types of data used in machine learning

- Numerical Data / Continuous Data / Signals / Vectors in $\mathbb{R}^D$

Data

# Types of data used in machine learning

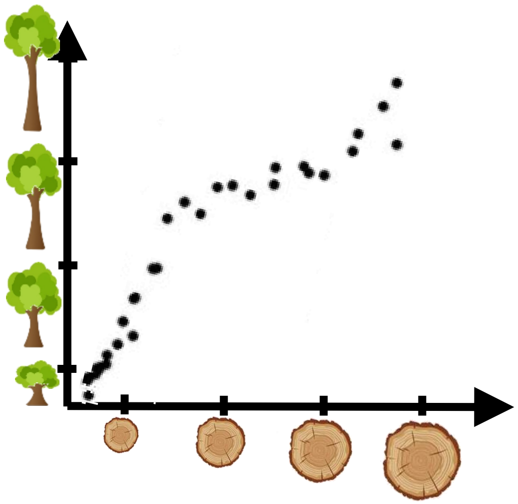
- Numerical Data / Continuous Data / Signals / Vectors in $\mathbb{R}^D$



Data

## Types of data used in machine learning

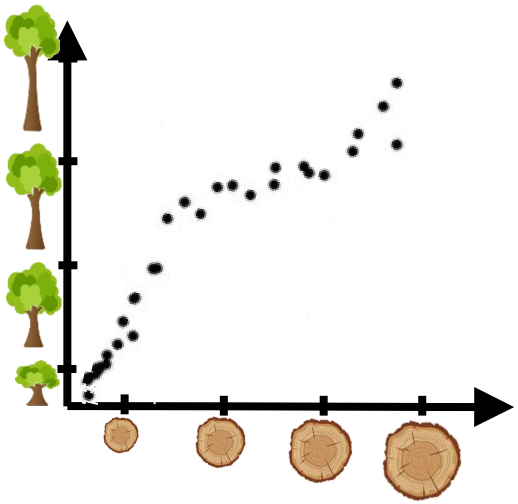
- Numerical Data / Continuous Data / Signals / Vectors in $\mathbb{R}^D$



Data

# Types of data used in machine learning

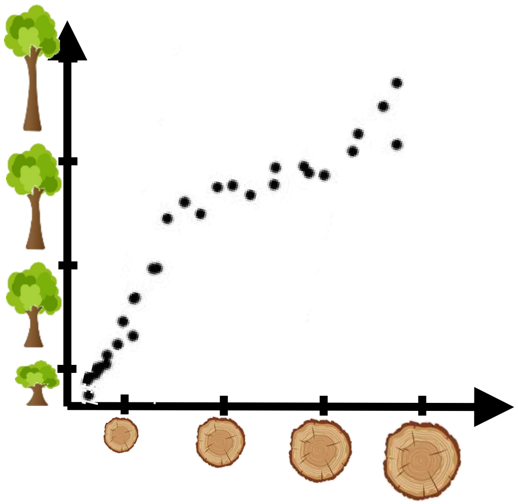
- Numerical Data / Continuous Data / Signals / Vectors in $\mathbb{R}^D$

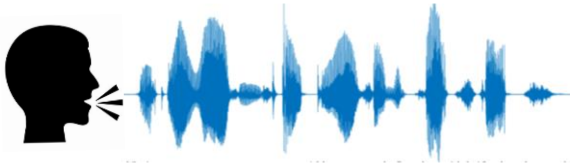


Data

# Types of data used in machine learning

- Numerical Data / Continuous Data / Signals / Vectors in $\mathbb{R}^D$

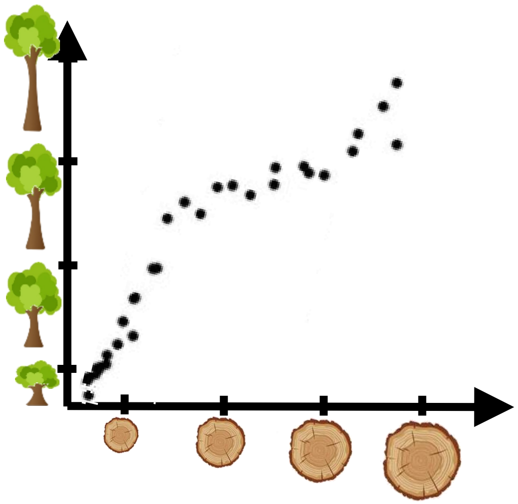


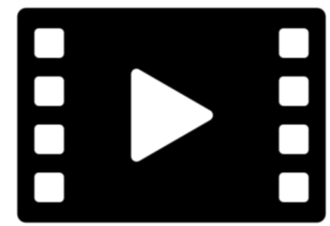


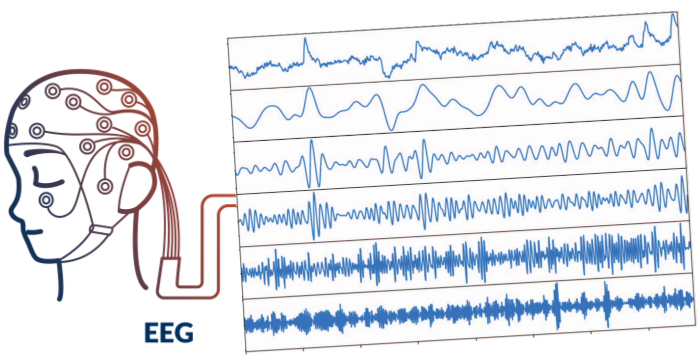
Data

# Types of data used in machine learning

- Numerical Data / Continuous Data / Signals / Vectors in $\mathbb{R}^D$



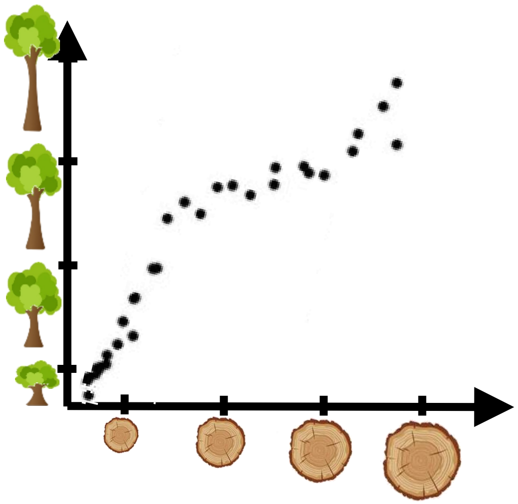


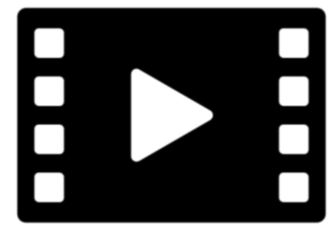


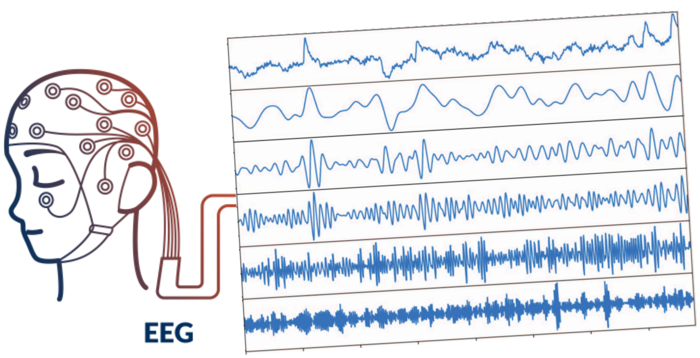
Data

# Types of data used in machine learning

- Numerical Data / Continuous Data / Signals / Vectors in $\mathbb{R}^D$








Data

# Types of data used in machine learning

- Categorical Data

ImageNet Labels

- airplane
- automobile
- bird
- cat
- deer
- dog
- frog
- horse
- ship
- truck
- …

Sex ∊ {male, female, other}
ZIP ∊ {67000, 75009, …}
Country ∊ {France, Spain, …}

Data

# Types of data used in machine learning

- Categorical Data

ImageNet Labels
- airplane
- automobile
- bird
- cat
- deer
- dog
- frog
- horse
- ship
- truck
- …

Sex ∈ {male, female, other}
ZIP ∈ {67000, 75009, …}
Country ∈ {France, Spain, …}

- Text Data

## Artificial intelligence

Article Talk

From Wikipedia, the free encyclopedia

**Artificial intelligence** (**AI**) is intelligence—perceiving, synthesizing, and inferring information—demonstrated by machines, as opposed to intelligence displayed by non-human animals and humans. Example tasks in which this is done include speech recognition, computer vision, translation between (natural) languages, as well as other mappings of inputs.

AI applications include advanced web search engines (e.g., Google Search), recommendation systems (used by YouTube, Amazon and Netflix), understanding human speech (such as Siri and Alexa), self-driving cars (e.g., Waymo), generative or creative tools (ChatGPT and AI art), automated decision-making and competing at the highest level in strategic game systems (such as chess and Go).[1]

Data

# Types of data used in machine learning

- Categorical Data

ImageNet Labels: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck, …

Sex ∊ {male, female, other}
ZIP ∊ {67000, 75009, …}
Country ∊ {France, Spain, …}

- Text Data

Artificial intelligence

Article Talk

From Wikipedia, the free encyclopedia

**Artificial intelligence** (**AI**) is intelligence—perceiving, synthesizing, and inferring information—demonstrated by machines, as opposed to intelligence displayed by non-human animals and humans. Example tasks in which this is done include speech recognition, computer vision, translation between (natural) languages, as well as other mappings of inputs.

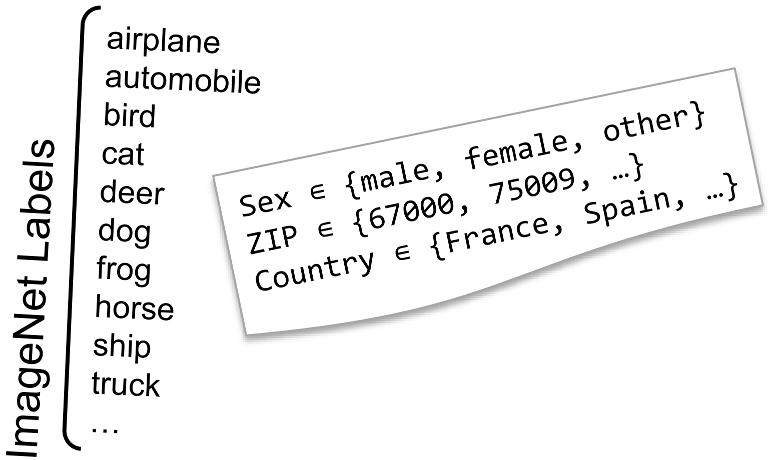
AI applications include advanced web search engines (e.g., Google Search), recommendation systems (used by YouTube, Amazon and Netflix), understanding human speech (such as Siri and Alexa), self-driving cars (e.g., Waymo), generative or creative tools (ChatGPT and AI art), automated decision-making and competing at the highest level in strategic game systems (such as chess and Go).[1]

- Heterogenous / Tabular Data

Data

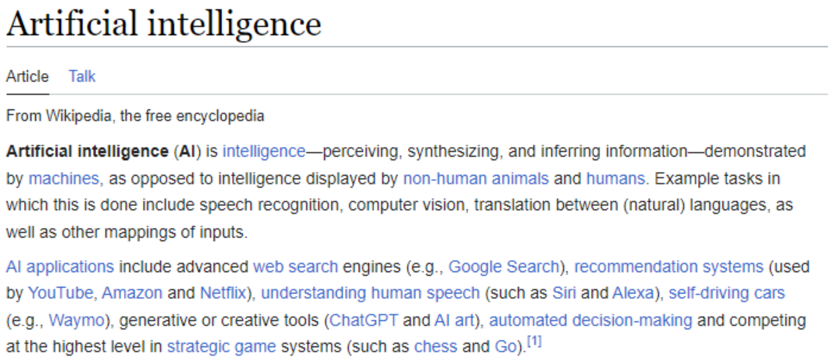
# Types of data used in machine learning

- Categorical Data




- Heterogenous / Tabular Data


- Text Data


Artificial intelligence

Article Talk

From Wikipedia, the free encyclopedia

**Artificial intelligence** (**AI**) is intelligence—perceiving, synthesizing, and inferring information—demonstrated by machines, as opposed to intelligence displayed by non-human animals and humans. Example tasks in which this is done include speech recognition, computer vision, translation between (natural) languages, as well as other mappings of inputs.

AI applications include advanced web search engines (e.g., Google Search), recommendation systems (used by YouTube, Amazon and Netflix), understanding human speech (such as Siri and Alexa), self-driving cars (e.g., Waymo), generative or creative tools (ChatGPT and AI art), automated decision-making and competing at the highest level in strategic game systems (such as chess and Go).[1]

- Time series

$$\cdots \to \boxed{x_{t-1}} \to \boxed{x_t} \to \boxed{x_{t+1}} \to \cdots$$

Data

# Types of data used in machine learning

- Categorical Data

ImageNet Labels: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck, …

```
Sex ∊ {male, female, other}
ZIP ∊ {67000, 75009, …}
Country ∊ {France, Spain, …}
```

- Heterogenous / Tabular Data


- Text Data

Artificial intelligence

Article Talk

From Wikipedia, the free encyclopedia

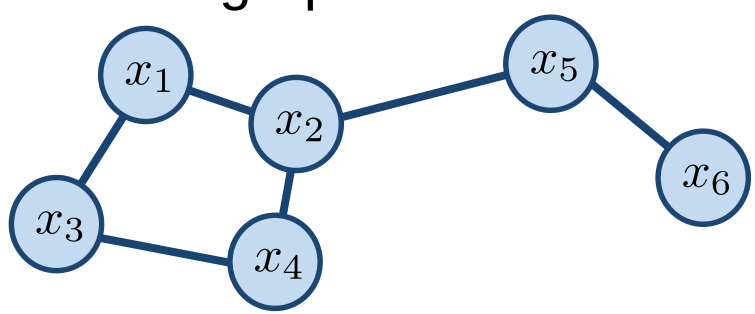
**Artificial intelligence** (**AI**) is intelligence—perceiving, synthesizing, and inferring information—demonstrated by machines, as opposed to intelligence displayed by non-human animals and humans. Example tasks in which this is done include speech recognition, computer vision, translation between (natural) languages, as well as other mappings of inputs.

AI applications include advanced web search engines (e.g., Google Search), recommendation systems (used by YouTube, Amazon and Netflix), understanding human speech (such as Siri and Alexa), self-driving cars (e.g., Waymo), generative or creative tools (ChatGPT and AI art), automated decision-making and competing at the highest level in strategic game systems (such as chess and Go).[1]

- Time series

$$\cdots \rightarrow \boxed{x_{t-1}} \rightarrow \boxed{x_t} \rightarrow \boxed{x_{t+1}} \rightarrow \cdots$$

- Data on graphs



# Pre-processing the data

Data

Data

# Pre-processing the data

- The individual pixels of an image or the samples of a waveform **correlate poorly** with explanatory variables of interest

Data

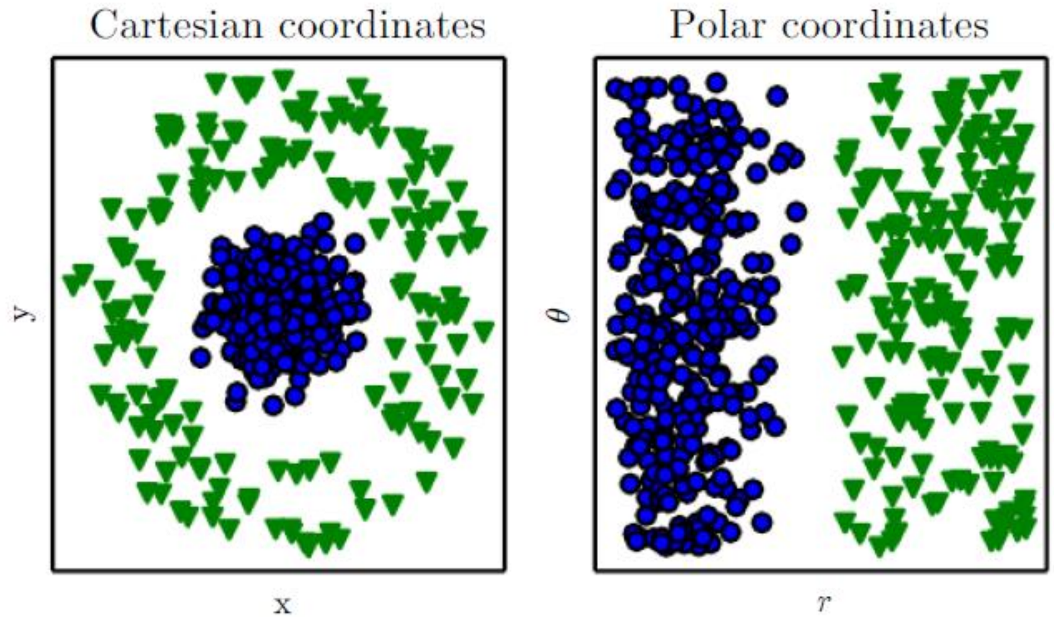
# Pre-processing the data

- The individual pixels of an image or the samples of a waveform **correlate poorly** with explanatory variables of interest
- *Conventional* machine learning methods define and compute **relevant features** before processing the *raw* data

## Pre-processing the data

Data

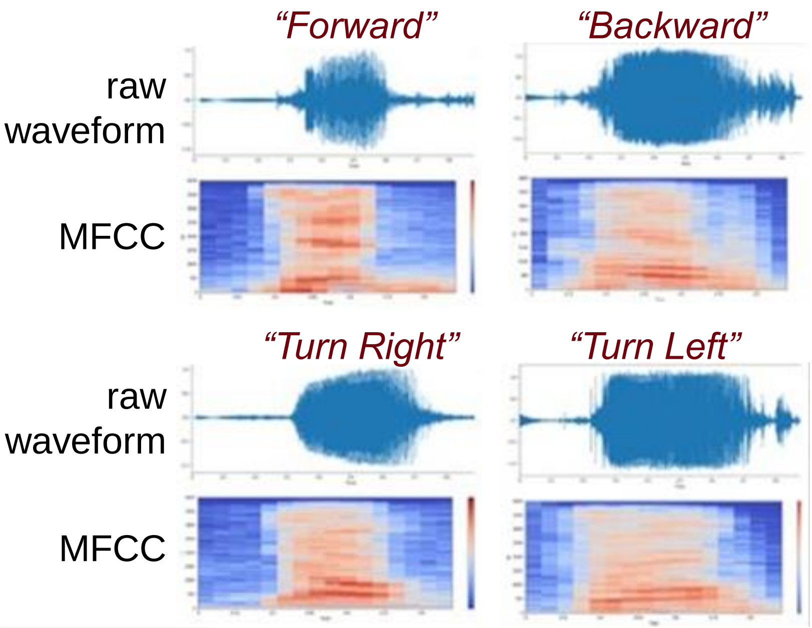
- The individual pixels of an image or the samples of a waveform **correlate poorly** with explanatory variables of interest
- *Conventional* machine learning methods define and compute **relevant features** before processing the *raw* data
- Example 1:



# Pre-processing the data

Data

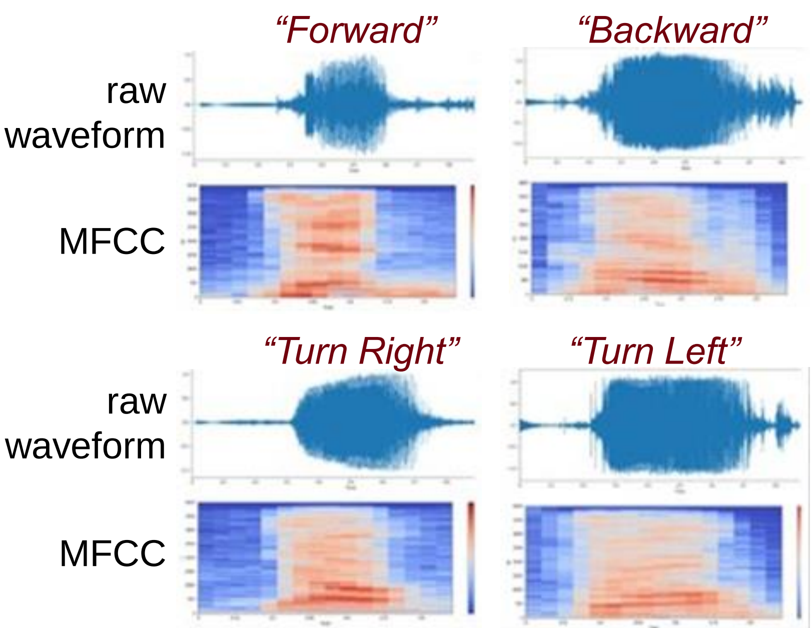
- The individual pixels of an image or the samples of a waveform **correlate poorly** with explanatory variables of interest
- *Conventional* machine learning methods define and compute **relevant features** before processing the *raw* data
- Example 2:



Data

# Pre-processing the data

- The individual pixels of an image or the samples of a waveform **correlate poorly** with explanatory variables of interest
- *Conventional* machine learning methods define and compute **relevant features** before processing the *raw* data
- Example 2:



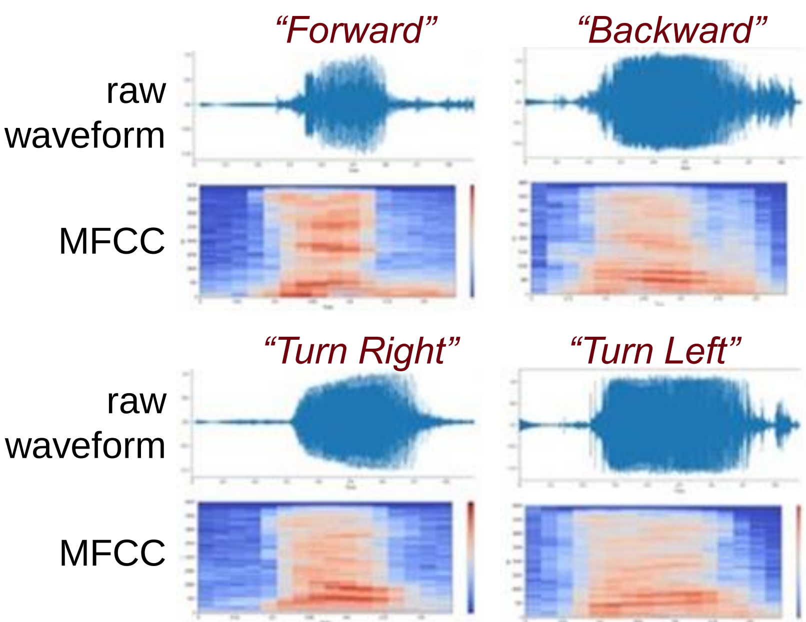

•The goals of features are:

1) **Disentangling** what's relevant
2) **Discarding** what isn't, i.e., build **invariance**

Data

# Pre-processing the data

- The individual pixels of an image or the samples of a waveform **correlate poorly** with explanatory variables of interest
- *Conventional* machine learning methods define and compute **relevant features** before processing the *raw* data
- Example 2:



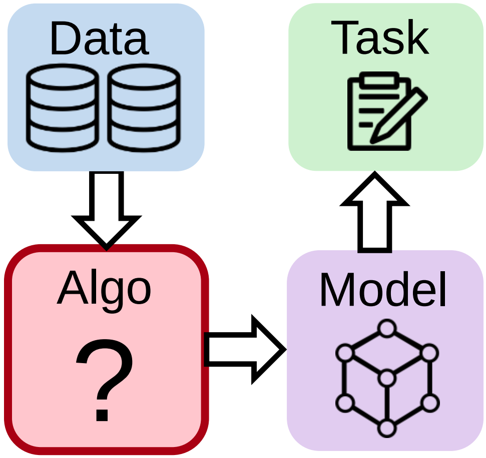

•The goals of features are:

1) **Disentangling** what's relevant
2) **Discarding** what isn't, i.e., build **invariance**

•Manually designing features, aka ***feature engineering***, can be hard for a given task

# Machine learning algorithms

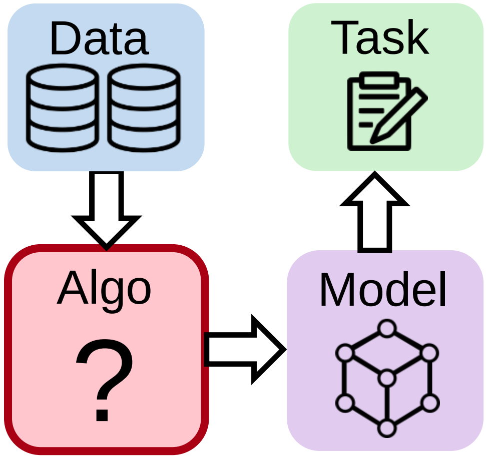
*How to make the program that makes the programs?*



# Machine learning algorithms

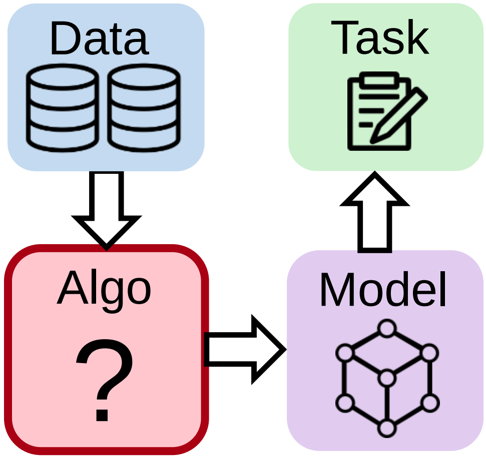
*How to make the program that makes the programs?*

- Search in the set of all python functions?



# Machine learning algorithms

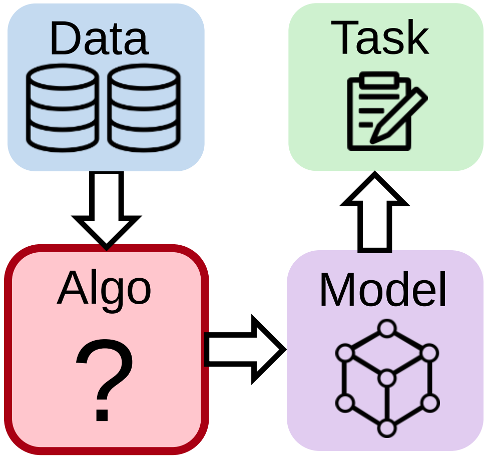
*How to make the program that makes the programs?*



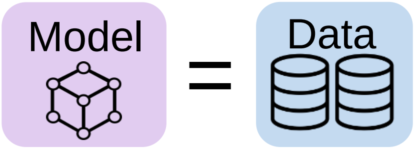

- Search in the set of all python functions?
  - 🚫 a gigantic combinatorial set

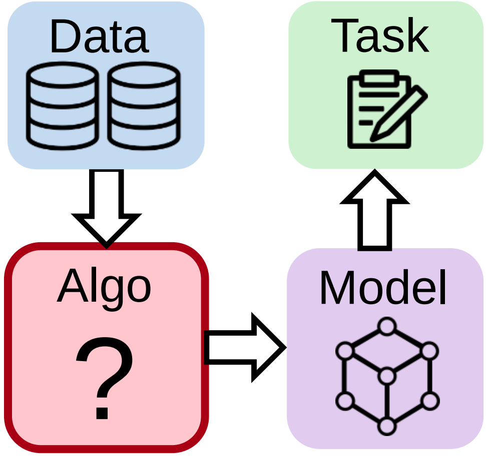
# Machine learning algorithms

*How to make the program that makes the programs?*




- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data

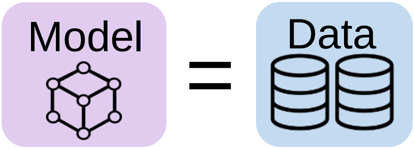





# Machine learning algorithms

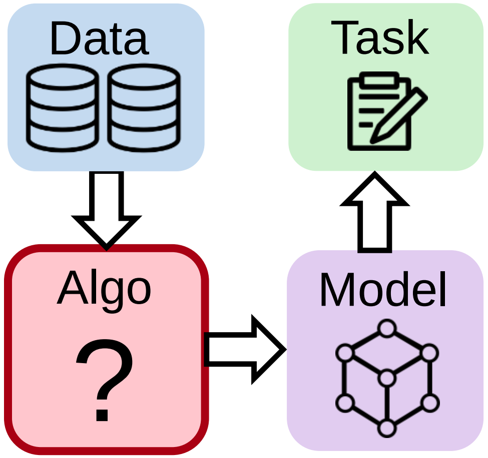
*How to make the program that makes the programs?*

- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data
  - *"Lazy Learning"*

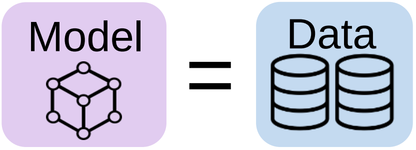


# Machine learning algorithms

*How to make the program that makes the programs?*

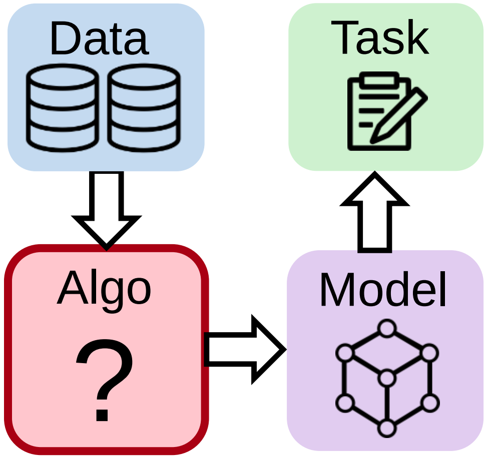



- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data



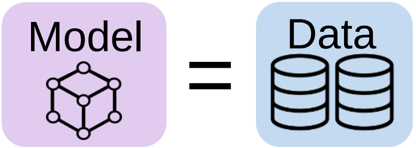

  - *“Lazy Learning”*
  - Ex: k-NN, look-up table, naive Bayes, case-based

# Machine learning algorithms




*How to make the program that makes the programs?*

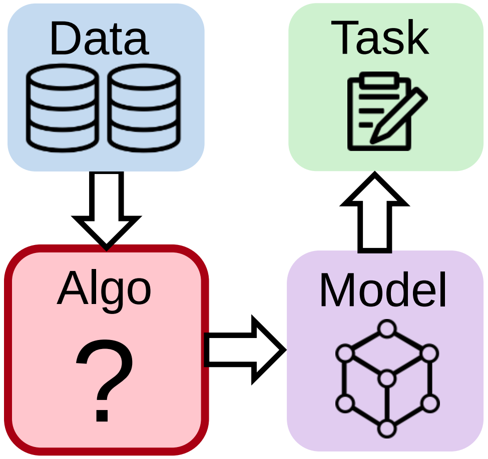
- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data



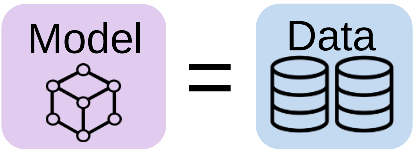

  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust

# Machine learning algorithms

*How to make the program that makes the programs?*



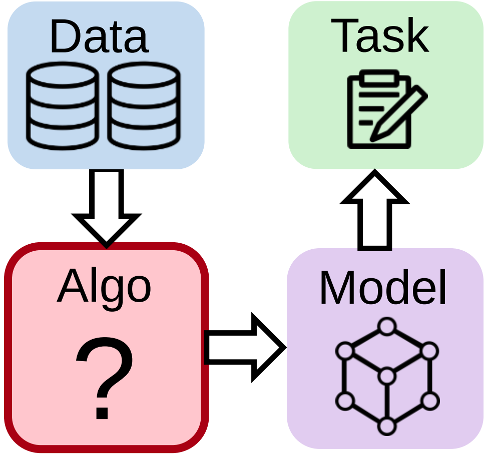

- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data



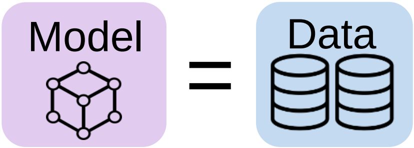

  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust
- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta\in\Theta}$

# Machine learning algorithms

*How to make the program that makes the programs?*




- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data

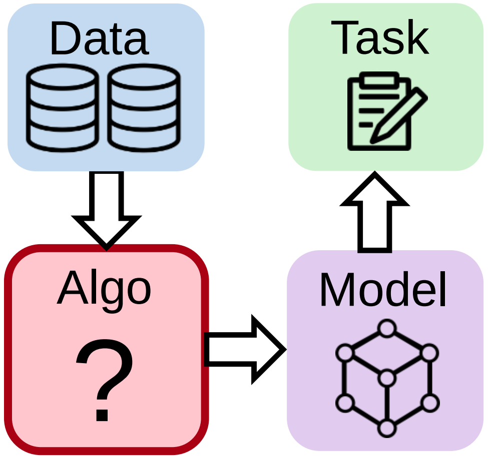



  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust
- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

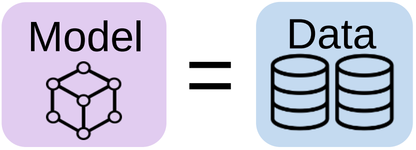
- $m_\theta =$ a jean
- $\theta =$ its (width, length)
- $\mathcal{F} =$ the shelves

# Machine learning algorithms

*How to make the program that makes the programs?*




- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data

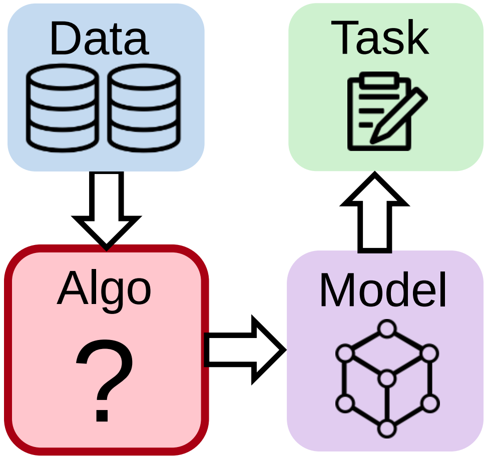



  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust
- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta\in\Theta}$

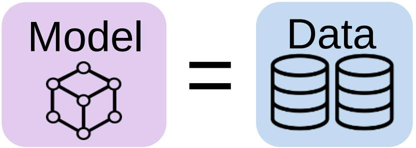
| $m$ is a function $f$ | $m$ is a prob. distribution $p$ |
|---|---|
| | |

# Machine learning algorithms

*How to make the program that makes the programs?*

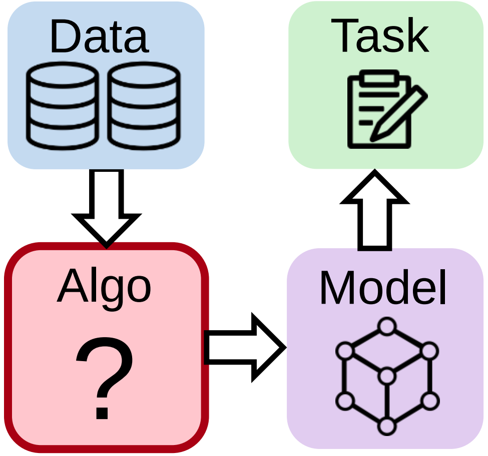



- Search in the set of all python functions?
  - 🚫 a gigantic combinatorial set
- Simply **memorize** the data




  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust
- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

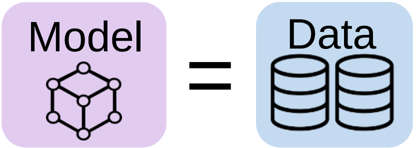
| $m$ is a function $f$ | $m$ is a prob. distribution $p$ |
|---|---|
| • $y = f_\theta(x)$ | |




# Machine learning algorithms

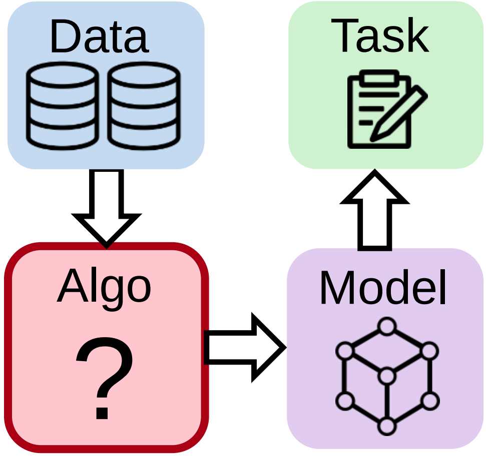
*How to make the program that makes the programs?*

- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data




  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust
- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

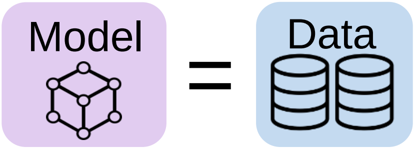
| $m$ is a function $f$ | $m$ is a prob. distribution $p$ |
|---|---|
| • $y = f_\theta(x)$<br>• $y_{\text{new}} = f_{\hat{\theta}}(x_{\text{new}})$ | |




# Machine learning algorithms

*How to make the program that makes the programs?*

- Search in the set of all python functions?
  - 🚫 a gigantic combinatorial set
- Simply **memorize** the data



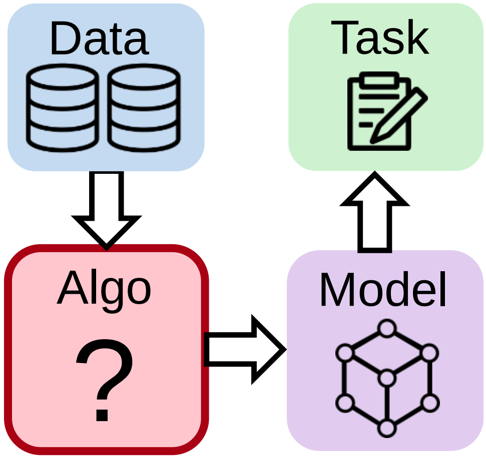

  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust

- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

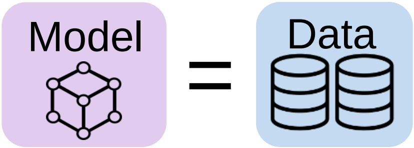
| $m$ is a function $f$ | $m$ is a prob. distribution $p$ |
|---|---|
| • $y = f_\theta(x)$ | • $p_\theta(X = x),\ p_\theta(X = x, Y = y)$ |
| • $y_{\text{new}} = f_{\hat{\theta}}(x_{\text{new}})$ | |

# Machine learning algorithms

*How to make the program that makes the programs?*




- Search in the set of all python functions?
  - 🚫 a gigantic combinatorial set
- Simply **memorize** the data



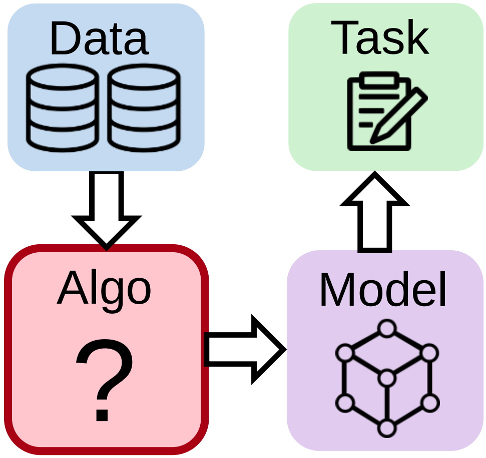

  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust

- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

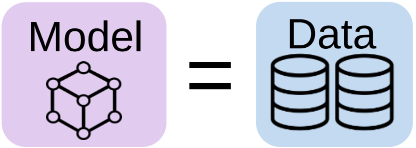
| $m$ is a function $f$ | $m$ is a prob. distribution $p$ |
|---|---|
| • $y = f_\theta(x)$ | • $p_\theta(X = x),\ p_\theta(X = x, Y = y)$ |
| • $y_{\text{new}} = f_{\hat{\theta}}(x_{\text{new}})$ | • $X \sim p_{\hat{\theta}},\ X \sim p_{\hat{\theta}}(\cdot \mid Y = y)$ |

# Machine learning algorithms

*How to make the program that makes the programs?*



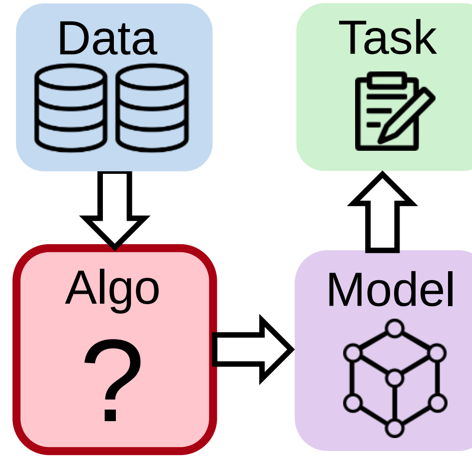

- Search in the set of all python functions?
  - a gigantic combinatorial set
- Simply **memorize** the data




  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust

- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

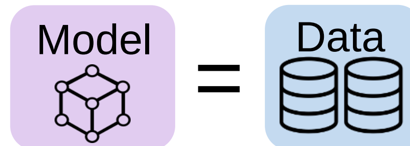
| $m$ is a function $f$ | $m$ is a prob. distribution $p$ |
|---|---|
| • $y = f_\theta(x)$ | • $p_\theta(X = x),\ p_\theta(X = x, Y = y)$ |
| • $y_{\text{new}} = f_{\hat{\theta}}(x_{\text{new}})$ | • $X \sim p_{\hat{\theta}},\ X \sim p_{\hat{\theta}}(\cdot \mid Y = y)$ |
| | • $\tilde{x} = \mathbb{E}_{p_{\hat{\theta}}(\cdot \mid y)}\{X\},\ \mathrm{var}(p_{\hat{\theta}}(\cdot \mid y))$ |




# Machine learning algorithms

*How to make the program that makes the programs?*

- Search in the set of all python functions?
  - 🚫 a gigantic combinatorial set
- Simply **memorize** the data



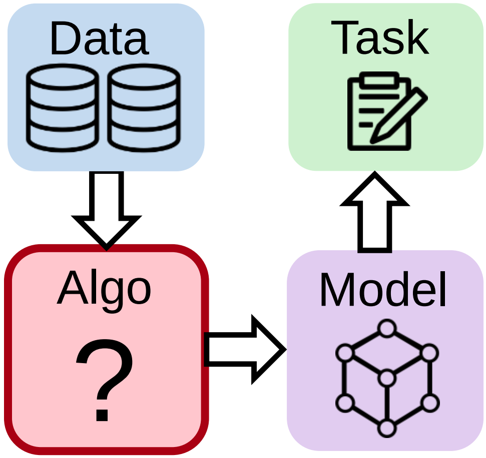

  - *"Lazy Learning"*
  - Ex: k-NN, look-up table, naive Bayes, case-based
  - Slow, large storage, non-robust
- **Model Fitting:**
  Find the *best* model within a **parameterized family** $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

| $m$ is a function $f$ | $m$ is a prob. distribution $p$ |
|---|---|
| • $y = f_\theta(x)$ | • $p_\theta(X = x),\ p_\theta(X = x, Y = y)$ |
| • $y_{\text{new}} = f_{\hat{\theta}}(x_{\text{new}})$ | • $X \sim p_{\hat{\theta}},\ X \sim p_{\hat{\theta}}(\cdot \vert Y = y)$ |
| | • $\tilde{x} = \mathbb{E}_{p_{\hat{\theta}}(\cdot \vert y)}\{X\},\ \text{var}(p_{\hat{\theta}}(\cdot \vert y))$ |

**Note:** It can be a mix of both

# Machine learning algorithms

*How to make the program that makes the programs?*

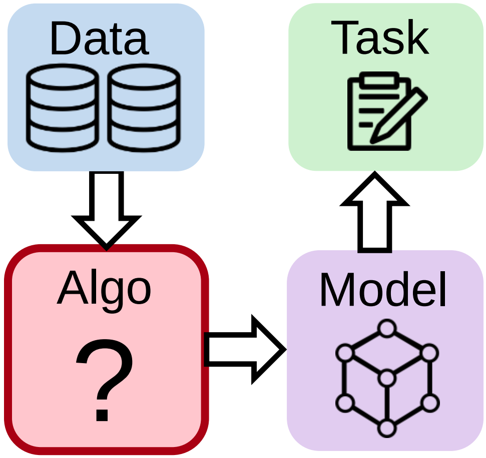



- The model is then found by minimizing a ***total loss*** / ***cost function*** $L$ over the set of parameters, for a given training **dataset** $\mathcal{T}$:

$$\hat{m} = m_{\hat{\theta}} \quad \text{where} \quad \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} L(m_\theta, \mathcal{T})$$

# Machine learning algorithms

*How to make the program that makes the programs?*



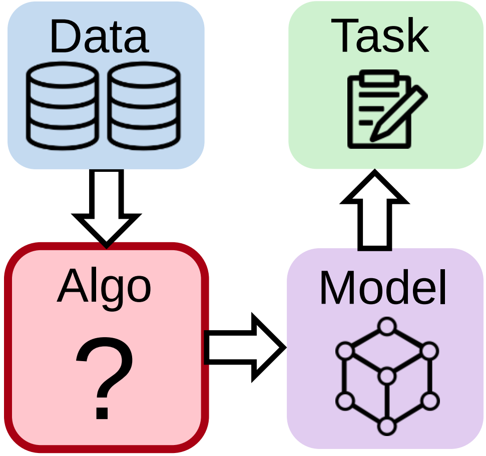

- The model is then found by minimizing a ***total loss*** / ***cost function*** $L$ over the set of parameters, for a given training **dataset** $\mathcal{T}$:

$$\hat{m} = m_{\hat{\theta}} \quad \text{where} \quad \hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \, L(m_\theta, \mathcal{T})$$

- The loss $L$ is designed based on the **task**, the **data**, and the chosen family of **models**. It measures the **fit** of $m_{\hat{\theta}}$ for these data and task.

# Machine learning algorithms

*How to make the program that makes the programs?*

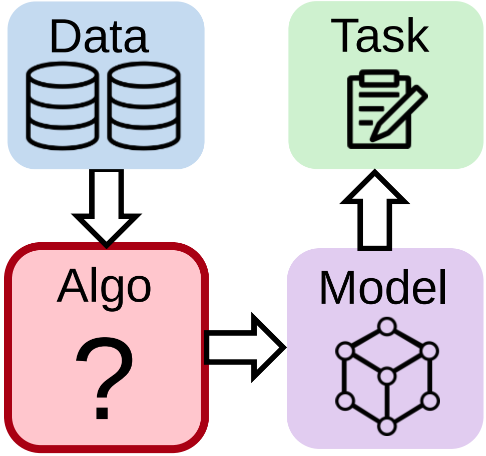



- The model is then found by minimizing a ***total loss / cost function*** $L$ over the set of parameters, for a given training **dataset** $\mathcal{T}$:

$$\hat{m} = m_{\hat{\theta}} \quad \text{where} \quad \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} L(m_\theta, \mathcal{T})$$

- The loss $L$ is designed based on the **task**, the **data**, and the chosen family of **models**. It measures the **fit** of $m_{\hat{\theta}}$ for these data and task.
- Most modern machine learning **algorithms** can be interpreted as minimizing a loss. They hence use **optimization**.

# Machine learning algorithms




*How to make the program that makes the programs?*

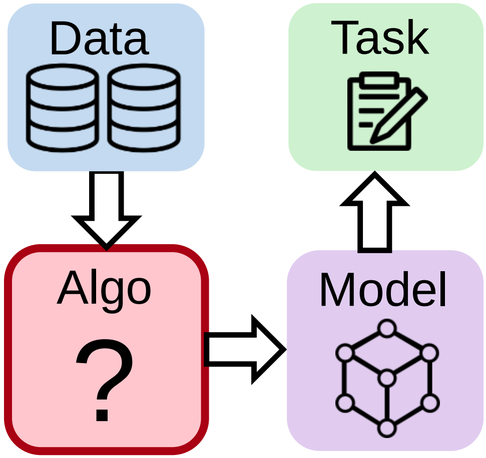
- The model is then found by minimizing a ***total loss*** / ***cost function*** $L$ over the set of parameters, for a given training **dataset** $\mathcal{T}$:

$$\hat{m} = m_{\hat{\theta}} \quad \text{where} \quad \hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \, L(m_\theta, \mathcal{T})$$

- The loss $L$ is designed based on the **task**, the **data**, and the chosen family of **models**. It measures the **fit** of $m_{\hat{\theta}}$ for these data and task.
- Most modern machine learning **algorithms** can be interpreted as minimizing a loss. They hence use **optimization**.
- The choice of an optimization method depends on the nature of the **loss** and of the **parameter set**.

# Machine learning algorithms

*How to make the program that makes the programs?*



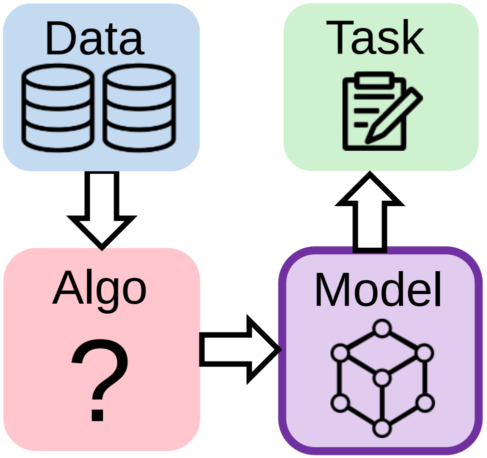

- The model is then found by minimizing a ***total loss / cost function*** $L$ over the set of parameters, for a given training **dataset** $\mathcal{T}$:

$$\hat{m} = m_{\hat{\theta}} \quad \text{where} \quad \hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}}\, L(m_\theta, \mathcal{T})$$

- The loss $L$ is designed based on the **task**, the **data**, and the chosen family of **models**. It measures the **fit** of $m_{\hat{\theta}}$ for these data and task.
- Most modern machine learning **algorithms** can be interpreted as minimizing a loss. They hence use **optimization**.
- The choice of an optimization method depends on the nature of the **loss** and of the **parameter set**.
- Optimization is a **huge** field. We will cover some of it in Chapter III.

# Machine learning algorithms

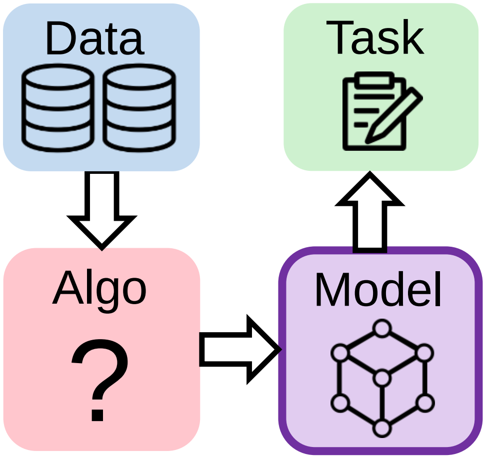
Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta\in\Theta}$




**Affine functions**

# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

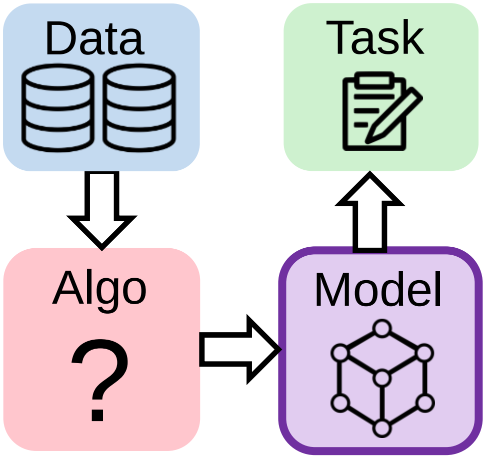



### Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$

# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

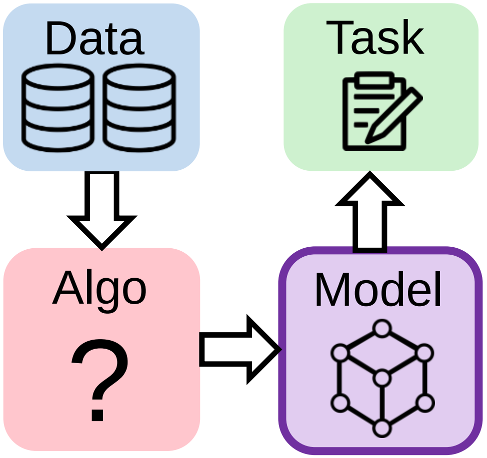



**Affine functions**

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$
- $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D \times d} \times \mathbb{R}^D$

# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

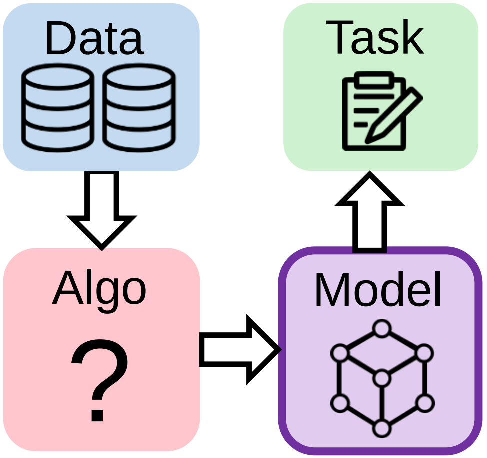



### Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$
- $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D \times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{T}} \|f_\theta(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$

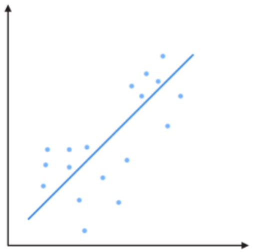
# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$



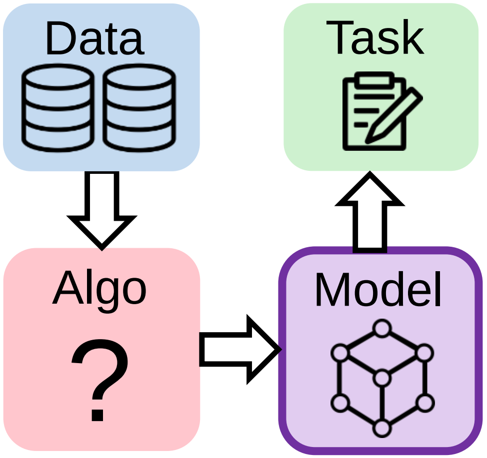

### Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$
- $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D \times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{T}} \|f_\theta(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$
- Special case: linear regression



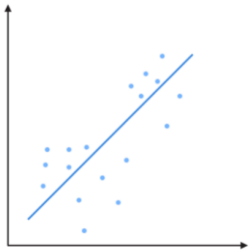
# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$




### Affine functions

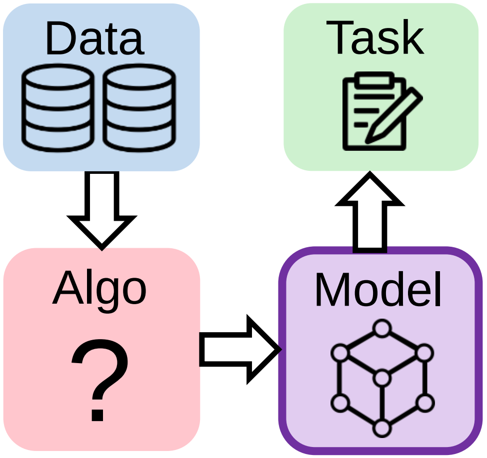
- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$
- $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D \times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{T}} \|f_\theta(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$
- Special case: linear regression



### Linear classifiers

- $f_\theta(\boldsymbol{x}) = \tau(\boldsymbol{w}^\top \boldsymbol{x} + b), \text{ where } \tau(u) = \begin{cases} 1 \text{ for } u \geq 0 \\ 0 \text{ for } u < 0 \end{cases}$

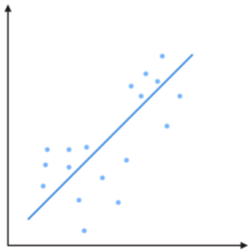
# Machine learning algorithms




Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

## Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$
- $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D \times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{T}} \|f_\theta(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$
- Special case: linear regression

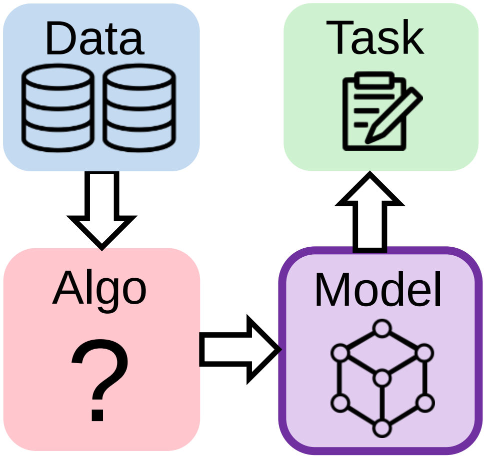


## Linear classifiers

- $f_\theta(\boldsymbol{x}) = \tau(\boldsymbol{w}^\top \boldsymbol{x} + b), \text{ where } \tau(u) = \begin{cases} 1 \text{ for } u \geq 0 \\ 0 \text{ for } u < 0 \end{cases}$
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$

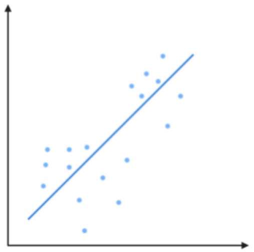
# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$



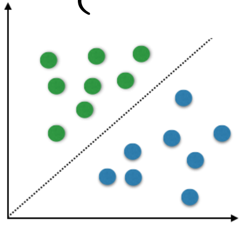

### Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$ • $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D \times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{T}} \|f_\theta(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$
- Special case: linear regression



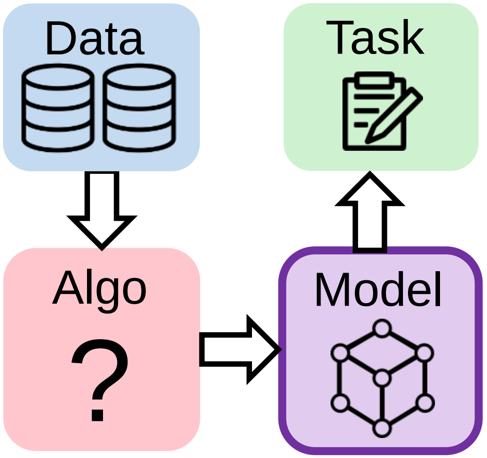
### Linear classifiers

- $f_\theta(\boldsymbol{x}) = \tau(\boldsymbol{w}^\top \boldsymbol{x} + b), \text{ where } \tau(u) = \begin{cases} 1 \text{ for } u \geq 0 \\ 0 \text{ for } u < 0 \end{cases}$
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$
- Typical loss: *cross entropy*



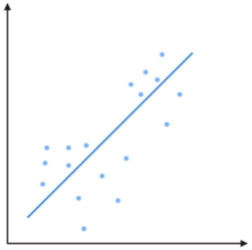
# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta\in\Theta}$



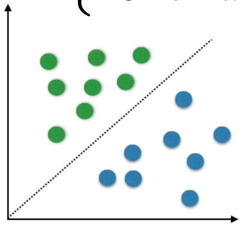

### Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$
- $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D\times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \|f_\theta(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$
- Special case: linear regression



### Linear classifiers

- $f_\theta(\boldsymbol{x}) = \tau(\boldsymbol{w}^\top\boldsymbol{x} + b)$, where $\tau(u) = \begin{cases} 1 \text{ for } u \geq 0 \\ 0 \text{ for } u < 0 \end{cases}$
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$
- Typical loss: *cross entropy*

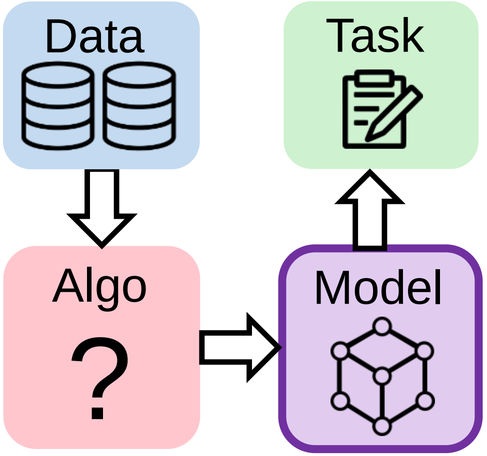


### Gaussian proba densities

- $p_\theta(\boldsymbol{X} = \boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$

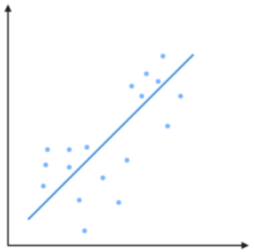
# Machine learning algorithms

Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$



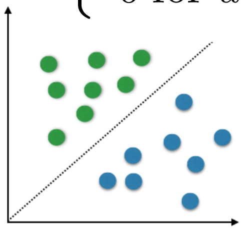

## Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$ • $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D\times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{T}} \|f_\theta(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$
- Special case: linear regression



## Linear classifiers

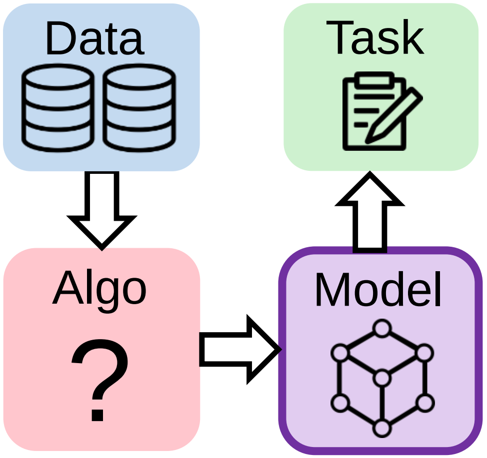
- $f_\theta(\boldsymbol{x}) = \tau(\boldsymbol{w}^\top \boldsymbol{x} + b)$, where $\tau(u) = \begin{cases} 1 \text{ for } u \geq 0 \\ 0 \text{ for } u < 0 \end{cases}$
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$
- Typical loss: *cross entropy*



## Gaussian proba densities

- $p_\theta(\boldsymbol{X} = \boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$
- $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathbb{R}^D \times \mathbb{R}^{D\times D}$

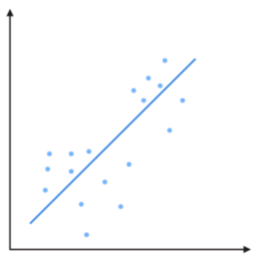
# Machine learning algorithms




Examples of **simple** families of models $\mathcal{F} = \{m_\theta\}_{\theta \in \Theta}$

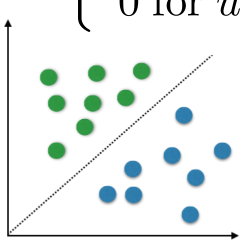
## Affine functions

- $f_\theta(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$
- $\theta = (\mathbf{A}, \boldsymbol{b}) \in \mathbb{R}^{D \times d} \times \mathbb{R}^D$
- Typical loss: $L(f_\theta, \mathcal{T}) = \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{T}} \| f_\theta(\boldsymbol{x}) - \boldsymbol{y} \|_2^2$
- Special case: linear regression



## Linear classifiers

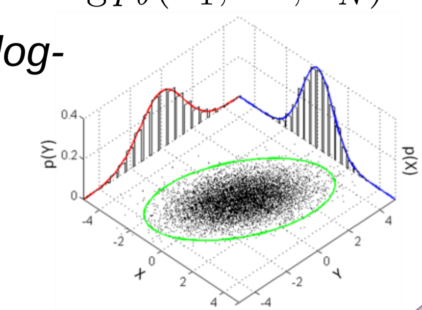
- $f_\theta(\boldsymbol{x}) = \tau(\boldsymbol{w}^\top \boldsymbol{x} + b), \text{ where } \tau(u) = \begin{cases} 1 \text{ for } u \geq 0 \\ 0 \text{ for } u < 0 \end{cases}$
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$
- Typical loss: *cross entropy*



## Gaussian proba densities

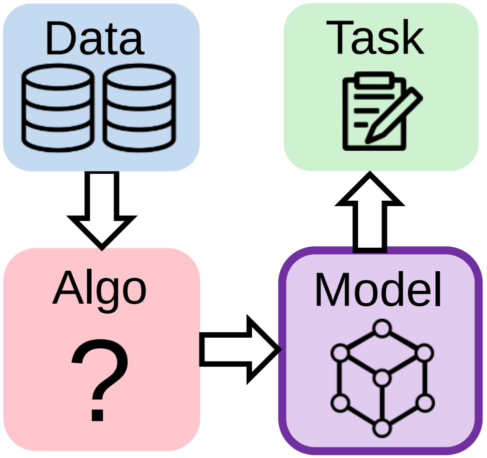
- $p_\theta(\boldsymbol{X} = \boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$
- $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathbb{R}^D \times \mathbb{R}^{D \times D}$
- $L(p_\theta, \mathcal{T}) = -\log p_\theta(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$

  *(negative log-likelihood)*



# Machine learning algorithms

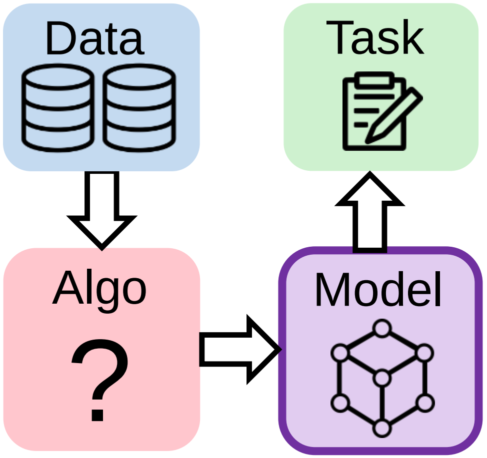
General approaches to **combine** models



# Machine learning algorithms

General approaches to **combine** models

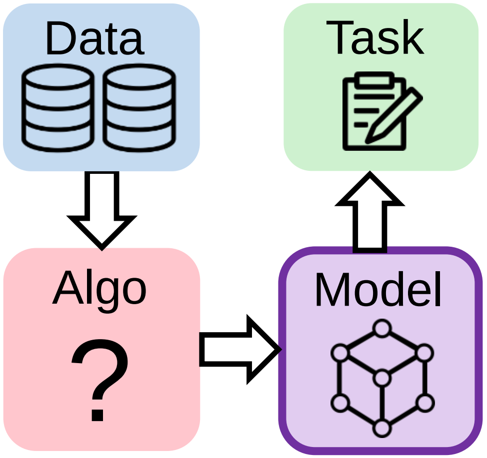
- Divide & Conquer



# Machine learning algorithms

General approaches to **combine** models

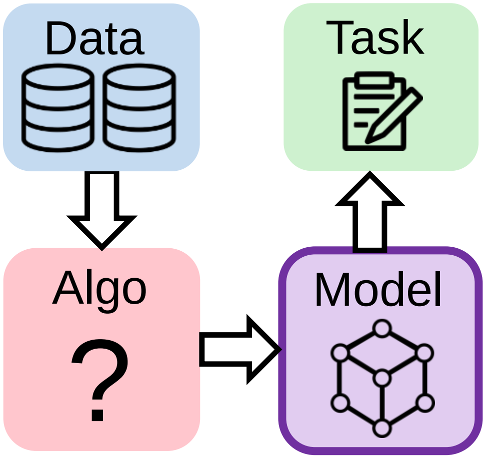
- Divide & Conquer
  - Partition the **input data** space



# Machine learning algorithms

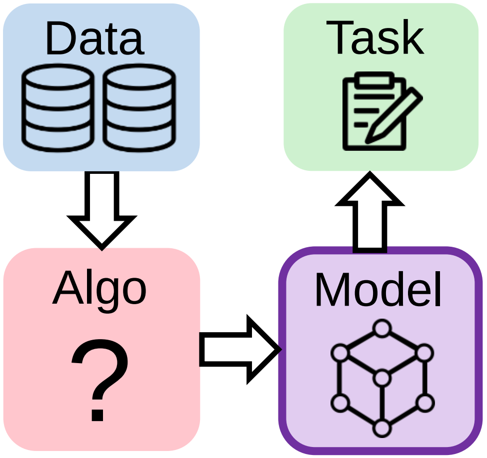
General approaches to **combine** models

- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert** **model** on each part

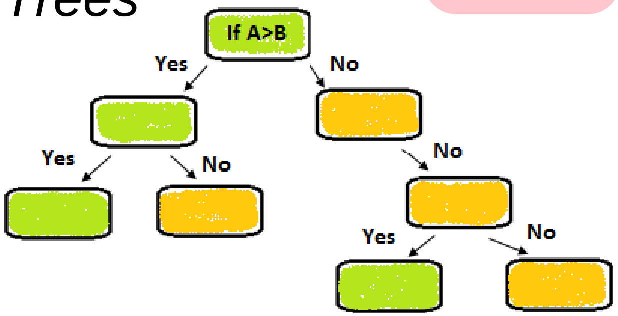


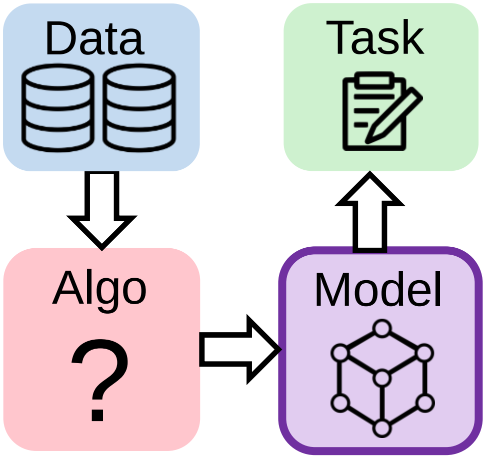
# Machine learning algorithms

General approaches to **combine** models




- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert model** on each part
  - Ex: *Mixtures of Experts, Decision Trees*

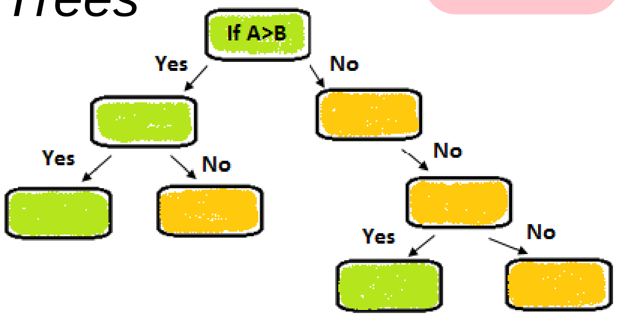




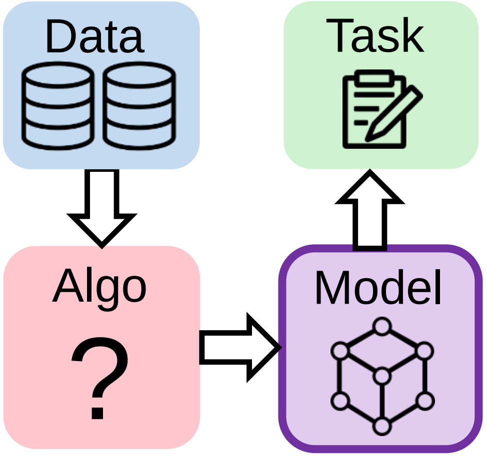

# Machine learning algorithms

## General approaches to **combine** models

- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert model** on each part
  - Ex: *Mixtures of Experts, Decision Trees*



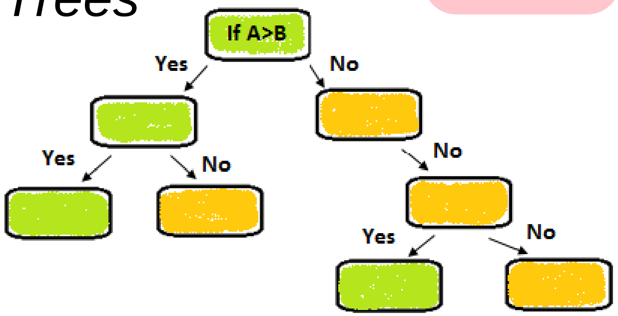

- Ensemble learning



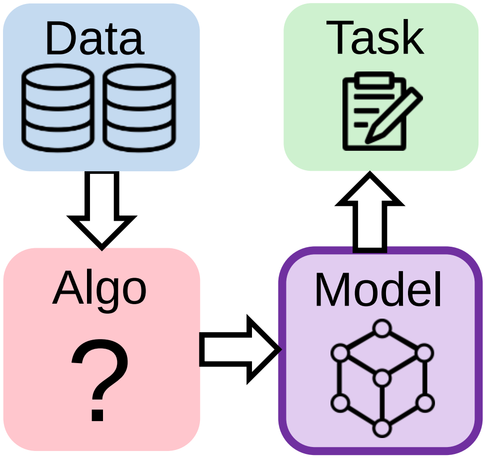

# Machine learning algorithms

General approaches to **combine** models

- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert model** on each part
  - Ex: *Mixtures of Experts, Decision Trees*



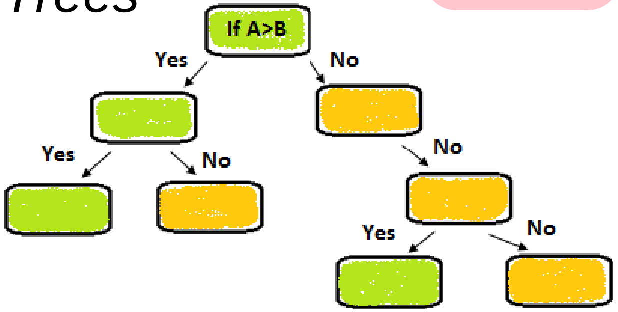

- Ensemble learning
  - Combine **several simple models** to get a better one




# Machine learning algorithms

General approaches to **combine** models

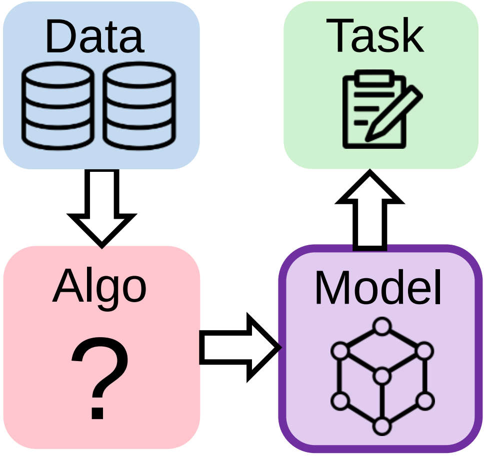
- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert model** on each part
  - Ex: *Mixtures of Experts, Decision Trees*



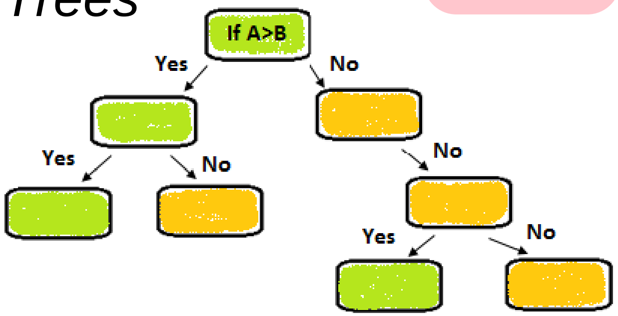

- Ensemble learning
  - Combine **several simple models** to get a better one
  - *Bagging* (or *Bootstrap*) = pool the output of several models (avg, vote)

# Machine learning algorithms

## General approaches to **combine** models



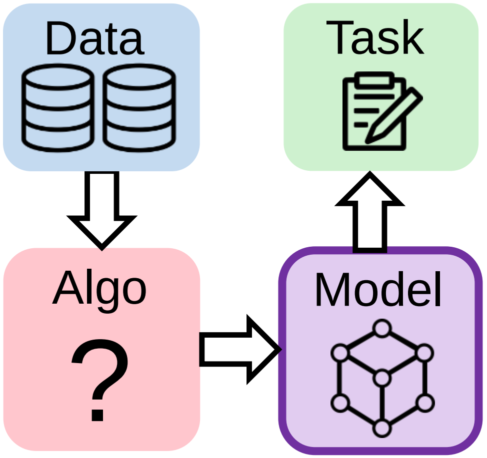

- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert model** on each part
  - Ex: *Mixtures of Experts, Decision Trees*



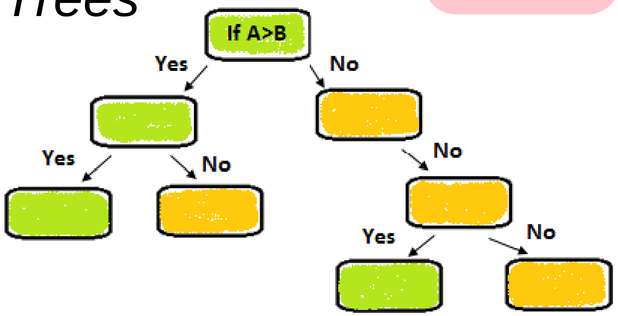

- Ensemble learning
  - Combine **several simple models** to get a better one
  - *Bagging* (or *Bootstrap*) = pool the output of several models (avg, vote)
  - *Boosting* = Sequentially train models, focusing on previously

# Machine learning algorithms

## General approaches to **combine** models



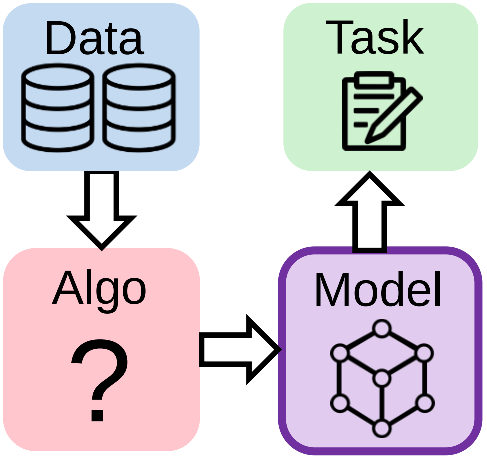

- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert model** on each part
  - Ex: *Mixtures of Experts, Decision Trees*



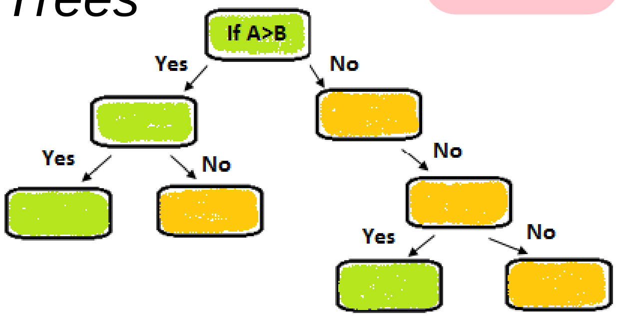

- Ensemble learning
  - Combine **several simple models** to get a better one
  - *Bagging* (or *Bootstrap*) = pool the output of several models (avg, vote)
  - *Boosting* = Sequentially train models, focusing on previously **misclassified** data

# Machine learning algorithms

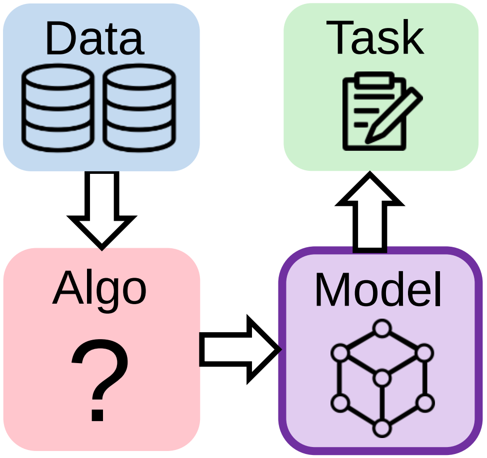
## General approaches to **combine** models




- Divide & Conquer
  - Partition the **input data** space
  - Train a **simple expert model** on each part
  - Ex: *Mixtures of Experts, Decision Trees*



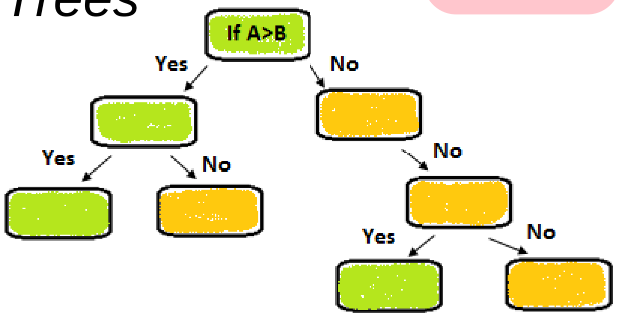

- Ensemble learning
  - Combine **several simple models** to get a better one
  - *Bagging* (or *Bootstrap*) = pool the output of several models (avg, vote)
  - *Boosting* = Sequentially train models, focusing on previously **misclassified** data
  - *Stacking* = Train a model to aggregate the output of multiple models



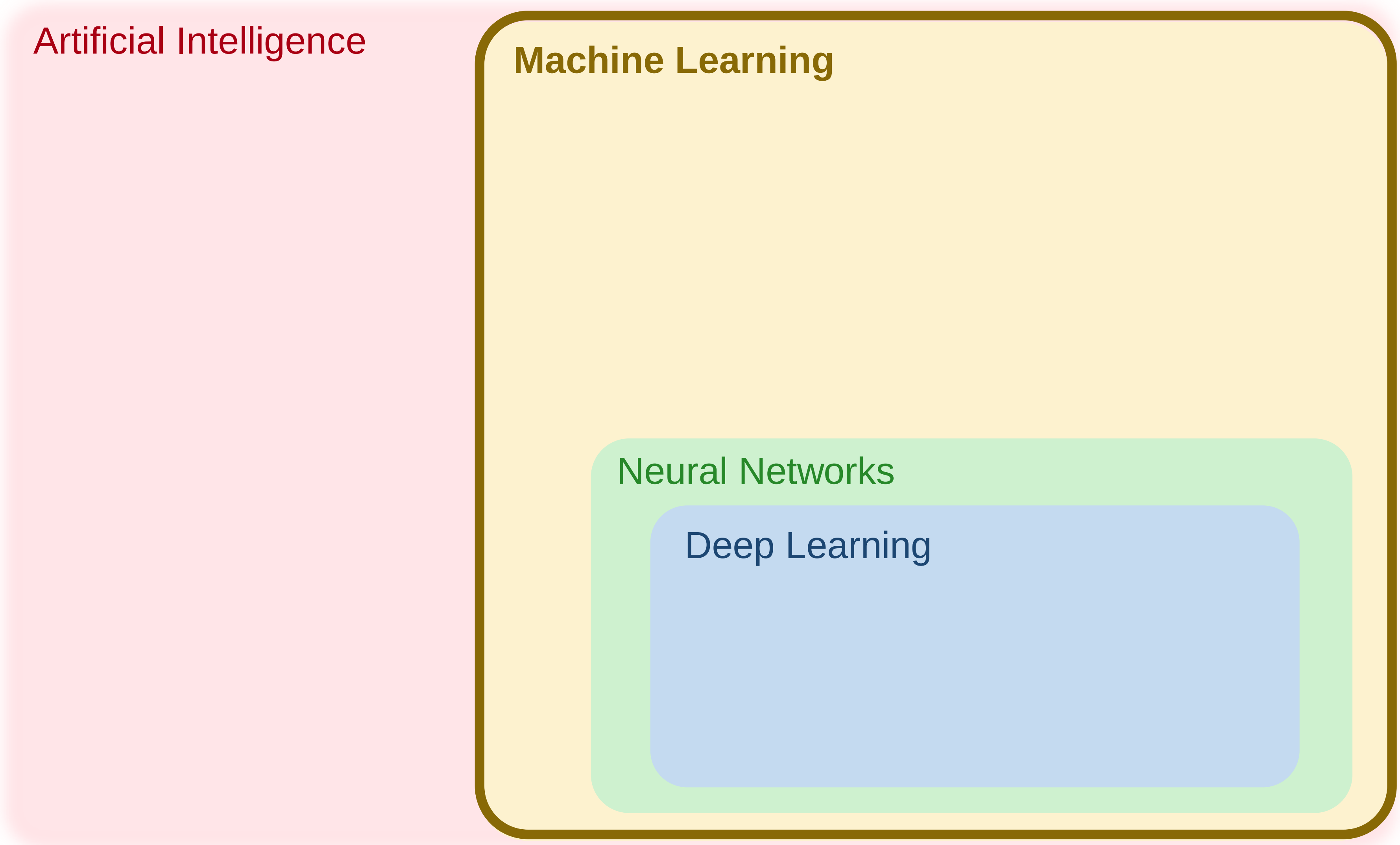

# Machine learning algorithms

## General approaches to **combine** models

- Divide & Conquer
    - Partition the **input data** space
    - Train a **simple expert model** on each part
    - Ex: *Mixtures of Experts, Decision Trees*




- Ensemble learning
    - Combine **several simple models** to get a better one
    - *Bagging* (or *Bootstrap*) = pool the output of several models (avg, vote)
    - *Boosting* = Sequentially train models, focusing on previously **misclassified** data
    - *Stacking* = Train a model to aggregate the output of multiple models
    - Ex: ***Random Forests*** = bagging of decision trees


Artificial Intelligence
Machine Learning
Neural Networks
Deep Learning

Artificial Intelligence

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**

Neural Networks

Deep Learning

Artificial Intelligence

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…

Neural Networks

Deep Learning

Artificial Intelligence

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models

Neural Networks

Deep Learning

Artificial Intelligence

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models
- *Conventional ML* = **feature engineering** + **combining** models from **small families**

Neural Networks

Deep Learning

Artificial Intelligence

Machine Learning

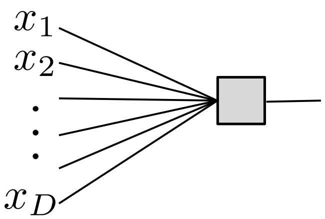
- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models
- *Conventional ML* = **feature engineering** + **combining** models from **small families**

**Neural Networks**

**Deep Learning**

# Artificial Neural Networks

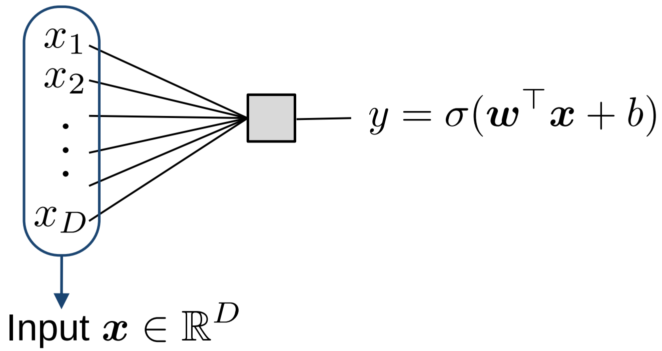
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




$$y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

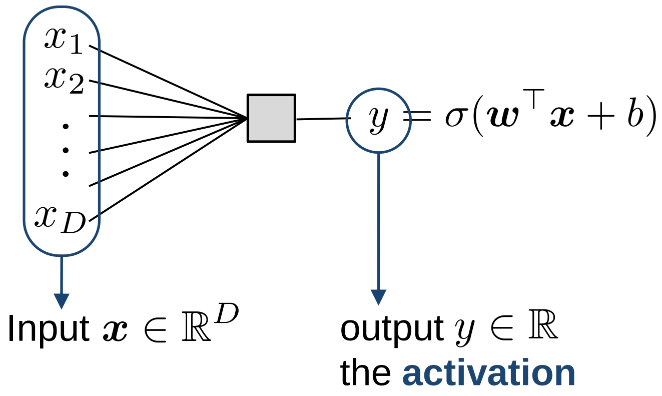
# Artificial Neural Networks

- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**



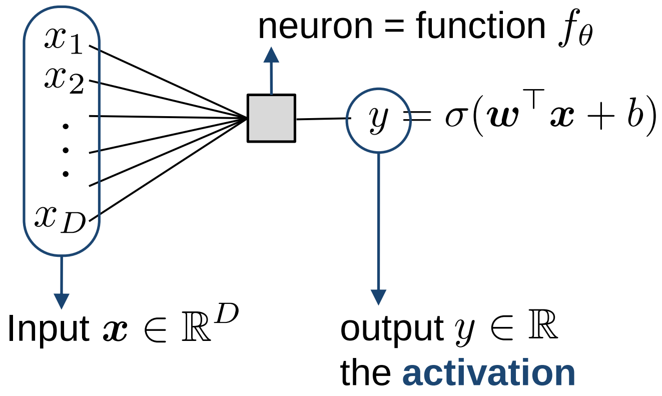
# Artificial Neural Networks

- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**



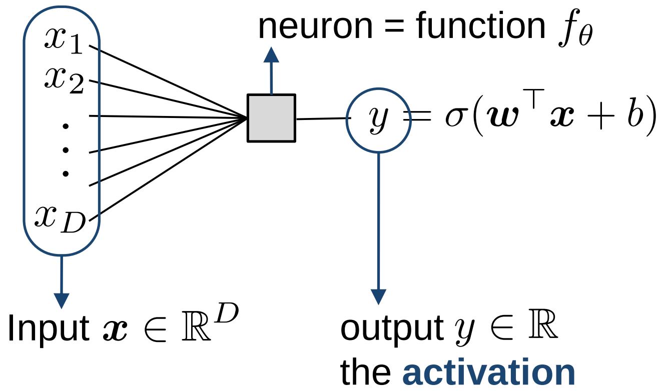
# Artificial Neural Networks

- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**



# Artificial Neural Networks

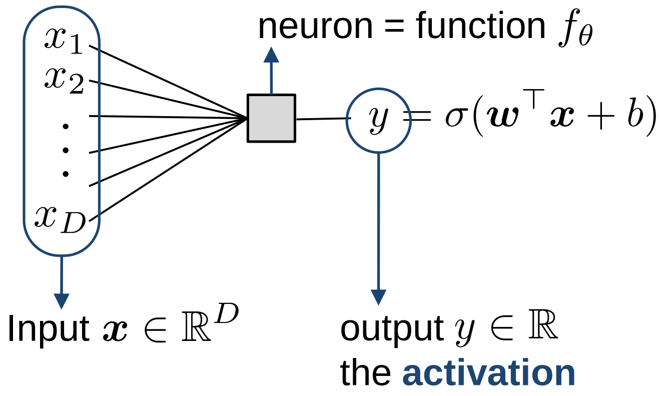
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***

# Artificial Neural Networks

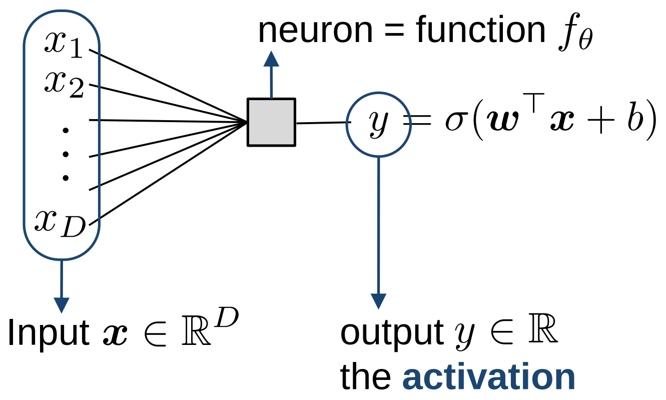
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \ldots, w_D]^\top \in \mathbb{R}^D$, the ***weights***

# Artificial Neural Networks

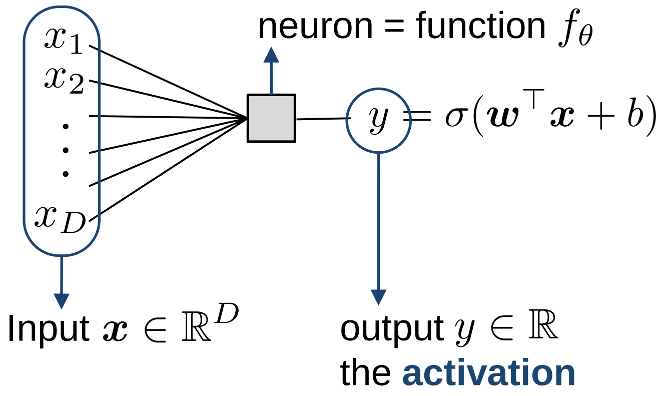
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \ldots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***

# Artificial Neural Networks

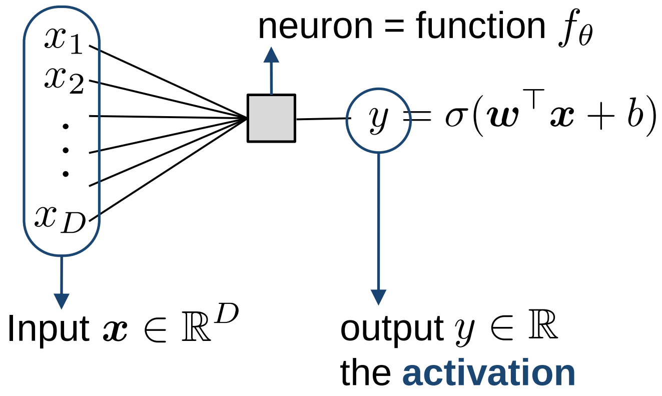
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \ldots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***

# Artificial Neural Networks

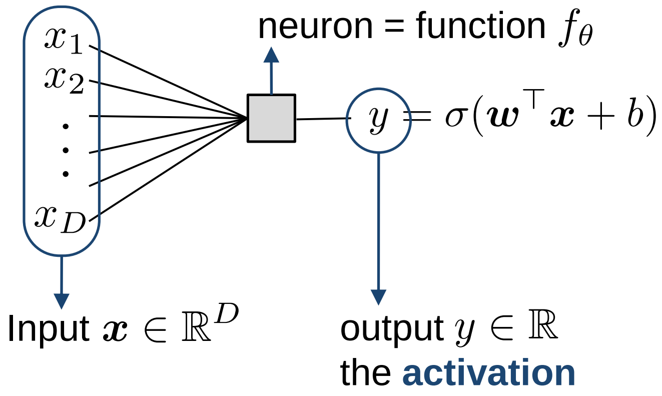
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \dots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$ , the ***parameters***

# Artificial Neural Networks

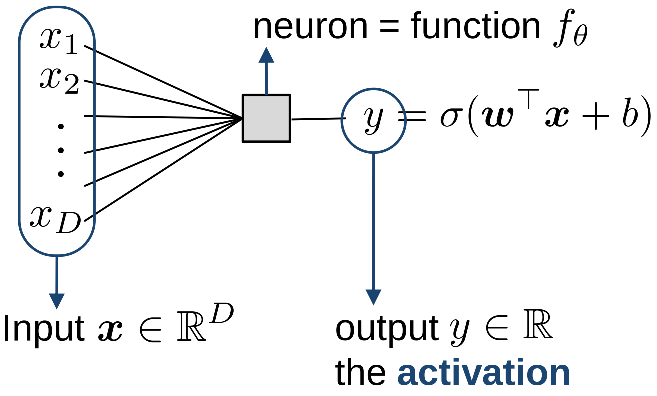
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \ldots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$ , the ***parameters***
- $\sigma$ , the ***activation function*** or ***non-linearity***.

# Artificial Neural Networks

- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**

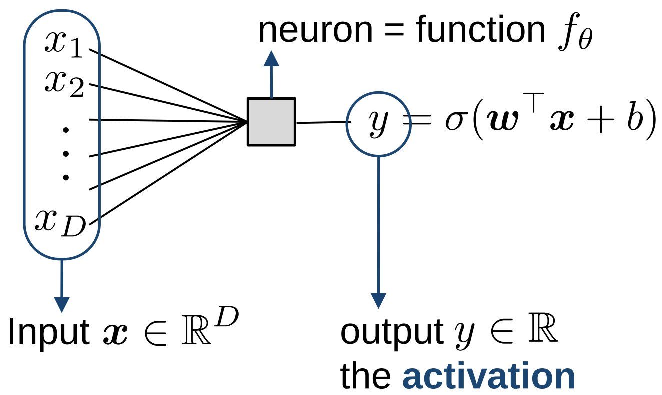



- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \ldots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$ , the ***parameters***
- $\sigma$ , the ***activation function*** or ***non-linearity***.

ex
- Sigmoid : $\sigma(x) = 1/(1 + e^{-x})$
- Rectified Linear Unit (ReLU) : $\sigma(x) = \max(0, x)$

# Artificial Neural Networks

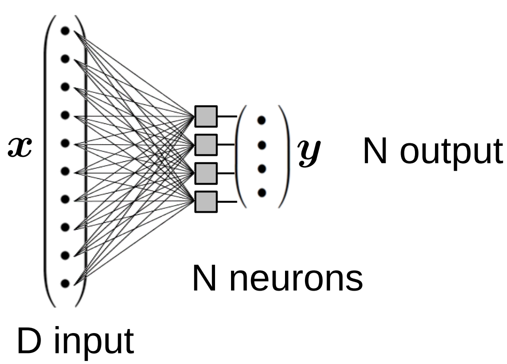
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \dots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$ , the ***parameters***
- $\sigma$ , the ***activation function*** or ***non-linearity***.

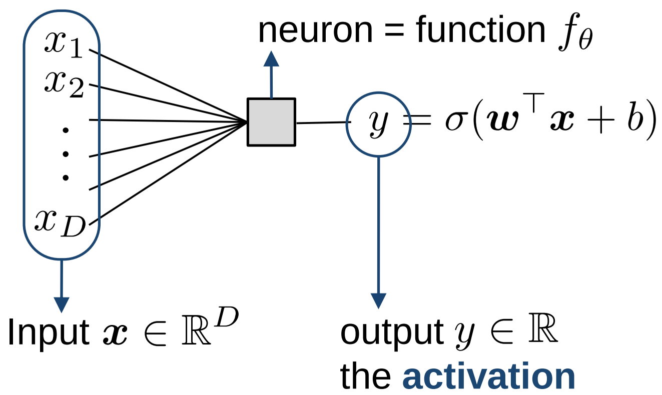
ex
- Sigmoid : $\sigma(x) = 1/(1 + e^{-x})$
- Rectified Linear Unit (ReLU) : $\sigma(x) = \max(0, x)$

- **Simple perceptron**: multiple neurons attending the same input



# Artificial Neural Networks

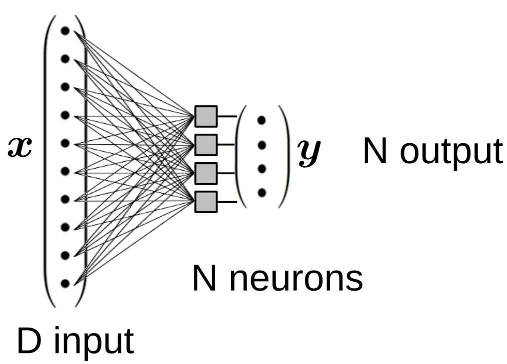
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \dots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$ , the ***parameters***
- $\sigma$ , the ***activation function*** or ***non-linearity***.

ex
- Sigmoid : $\sigma(x) = 1/(1+e^{-x})$
- Rectified Linear Unit (ReLU) : $\sigma(x) = \max(0, x)$

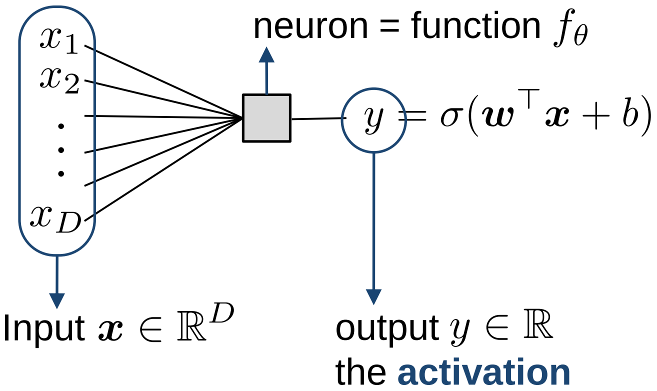
- **Simple perceptron**: multiple neurons attending the same input




- We have: $y_n = \sigma(\boldsymbol{w}_n^\top \boldsymbol{x} + b_n)$

# Artificial Neural Networks

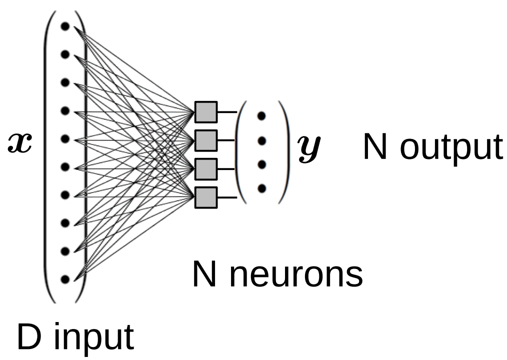
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \dots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$ , the ***parameters***
- $\sigma$ , the ***activation function*** or ***non-linearity***.

ex
- Sigmoid : $\sigma(x) = 1/(1 + e^{-x})$
- Rectified Linear Unit (ReLU) : $\sigma(x) = \max(0, x)$

- **Simple perceptron**: multiple neurons attending the same input

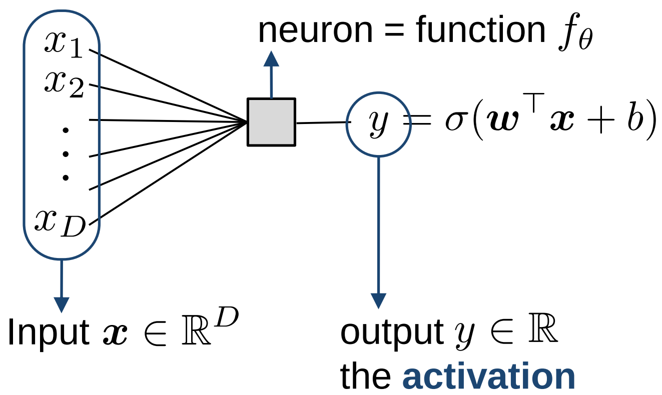



- We have: $y_n = \sigma(\boldsymbol{w}_n^\top \boldsymbol{x} + b_n)$
- Or in matrix form: $\boldsymbol{y} = \sigma(\mathbf{W}\boldsymbol{x} + \boldsymbol{b})$ , where

$$\mathbf{W} = \begin{bmatrix} \boldsymbol{w}_1^\top \\ \boldsymbol{w}_2^\top \\ \vdots \\ \boldsymbol{w}_N^\top \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,D} \\ w_{2,1} & w_{2,2} & \dots & w_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N,1} & w_{N,2} & \dots & w_{N,D} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}, \quad \theta = (\mathbf{W}, \boldsymbol{b})$$

# Artificial Neural Networks

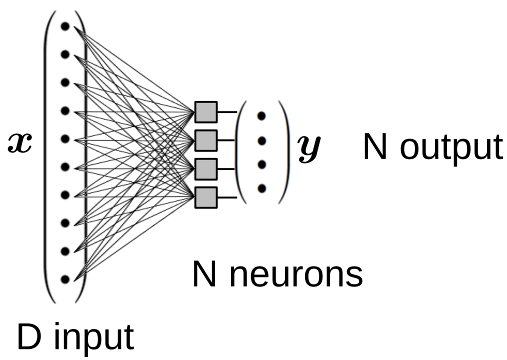
- **Artificial Neuron** = a multiple-input, single-output, **parametric, nonlinear function**




- $\boldsymbol{w}^\top \boldsymbol{x} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \sum_{d=1}^{D} w_d x_d$ , the ***dot product***
- $\boldsymbol{w} = [w_1, \dots, w_D]^\top \in \mathbb{R}^D$, the ***weights***
- $b \in \mathbb{R}$ , the ***bias***
- $\boldsymbol{w}^\top \boldsymbol{x} + b$ : the ***pre-activation***
- $\theta = (\boldsymbol{w}, b) \in \mathbb{R}^D \times \mathbb{R}$ , the ***parameters***
- $\sigma$ , the ***activation function*** or ***non-linearity***.

ex
- Sigmoid : $\sigma(x) = 1/(1 + e^{-x})$
- Rectified Linear Unit (ReLU) : $\sigma(x) = \max(0, x)$

- **Simple perceptron**: multiple neurons attending the same input



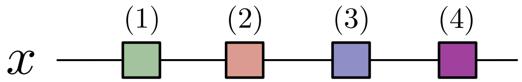

- We have: $y_n = \sigma(\boldsymbol{w}_n^\top \boldsymbol{x} + b_n)$
- Or in matrix form: $\boldsymbol{y} = \sigma(\mathbf{W}\boldsymbol{x} + \boldsymbol{b})$ , where

$$\mathbf{W} = \begin{bmatrix} \boldsymbol{w}_1^\top \\ \boldsymbol{w}_2^\top \\ \vdots \\ \boldsymbol{w}_N^\top \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,D} \\ w_{2,1} & w_{2,2} & \dots & w_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N,1} & w_{N,2} & \dots & w_{N,D} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}, \quad \theta = (\mathbf{W}, \boldsymbol{b}) \in \mathbb{R}^{N \times D} \times \mathbb{R}^N$$

- Note: In this notation, $\sigma$ is applied **elementwise** to a vector

# Artificial Neural Networks

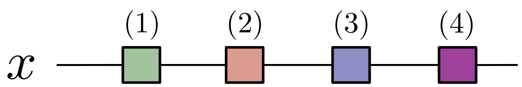
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

# Artificial Neural Networks

- Neurons can be **chained** together

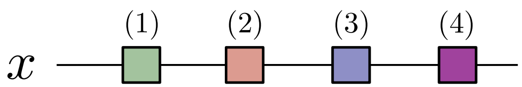



$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

# Artificial Neural Networks

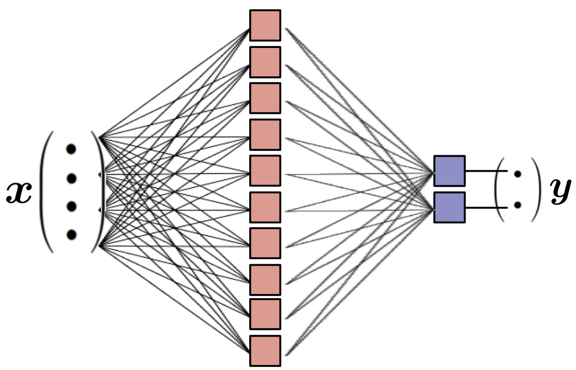
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

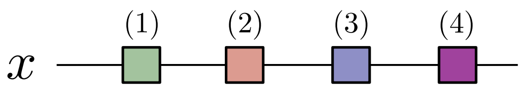
$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**



# Artificial Neural Networks

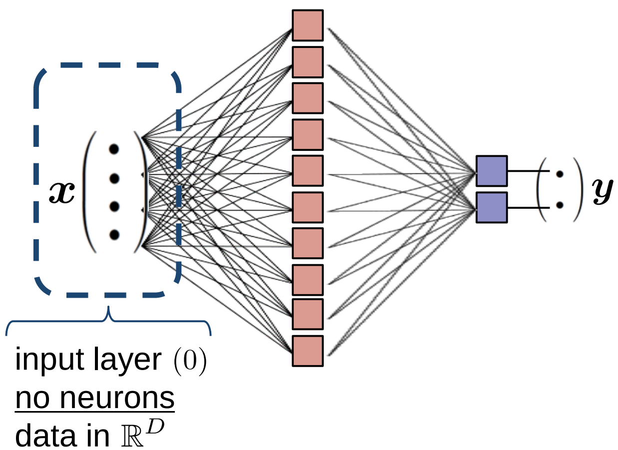
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

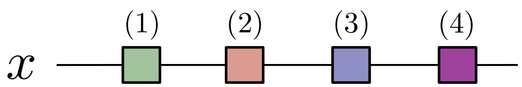
$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**



# Artificial Neural Networks

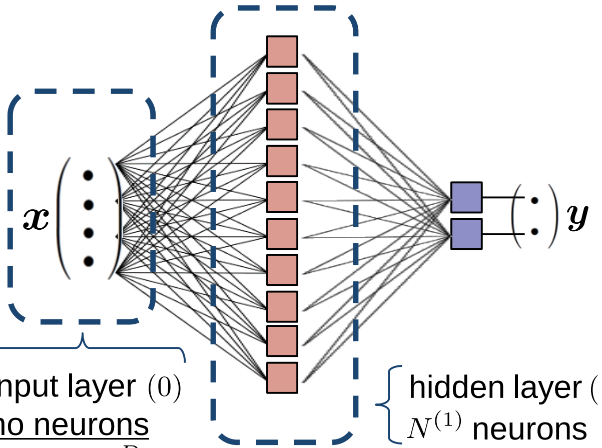
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**

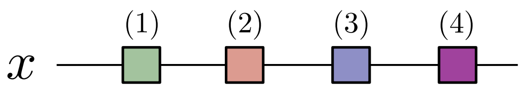



input layer $(0)$
<u>no neurons</u>
data in $\mathbb{R}^D$

hidden layer $(1)$
$N^{(1)}$ neurons

# Artificial Neural Networks

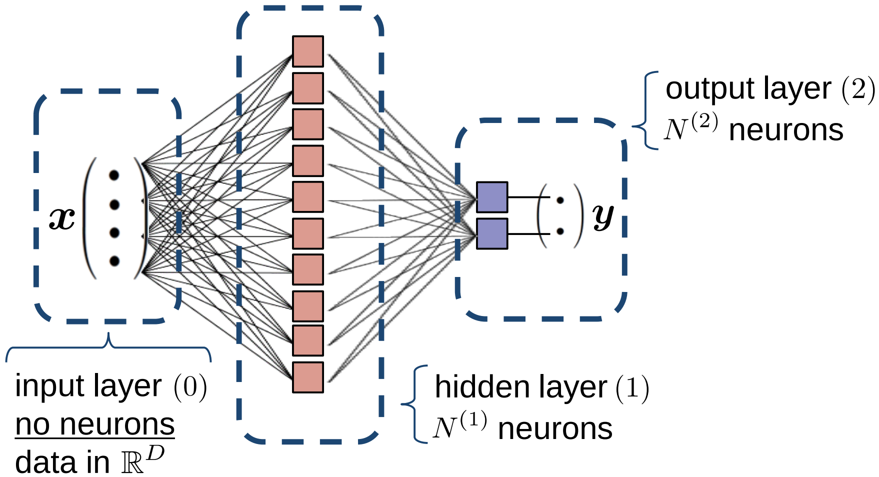
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

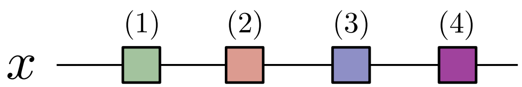
$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**



# Artificial Neural Networks

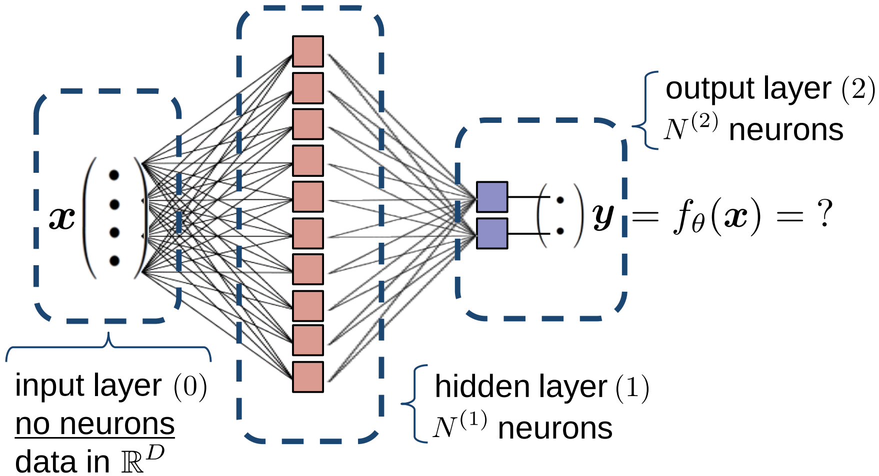
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

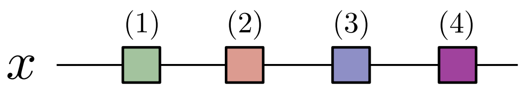
$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**



# Artificial Neural Networks

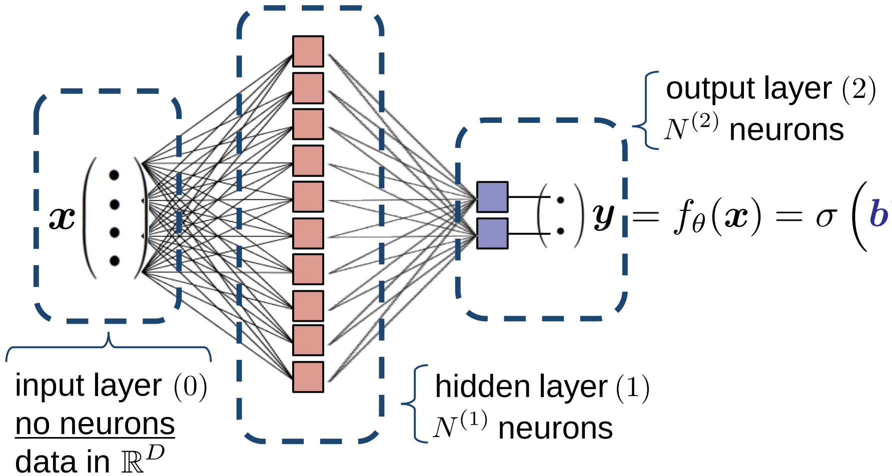
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

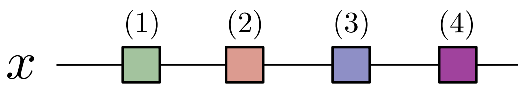
- Perceptron with a **single hidden layer**




$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}^{(2)} + \mathbf{W}^{(2)} \cdot \sigma\left(\boldsymbol{b}^{(1)} + \mathbf{W}^{(1)} \cdot \boldsymbol{x}\right)\right)$$

# Artificial Neural Networks

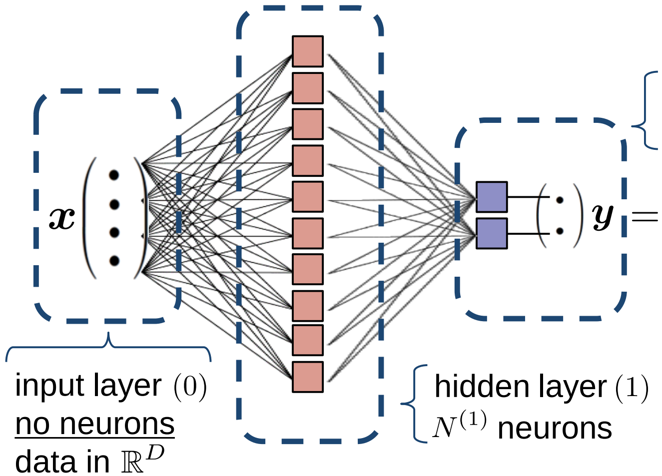
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**

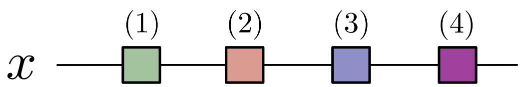



$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}^{(2)} + \mathbf{W}^{(2)} \cdot \sigma\left(\boldsymbol{b}^{(1)} + \mathbf{W}^{(1)} \cdot \boldsymbol{x}\right)\right)$$

$$\theta = \{\boldsymbol{b}^{(1)}, \mathbf{W}^{(1)}, \boldsymbol{b}^{(2)}, \mathbf{W}^{(2)}\} \in \mathbb{R}^{N^{(1)}} \times \mathbb{R}^{N^{(1)} \times D} \times \mathbb{R}^{N^{(2)}} \times \mathbb{R}^{N^{(2)} \times N^{(1)}}$$

# Artificial Neural Networks

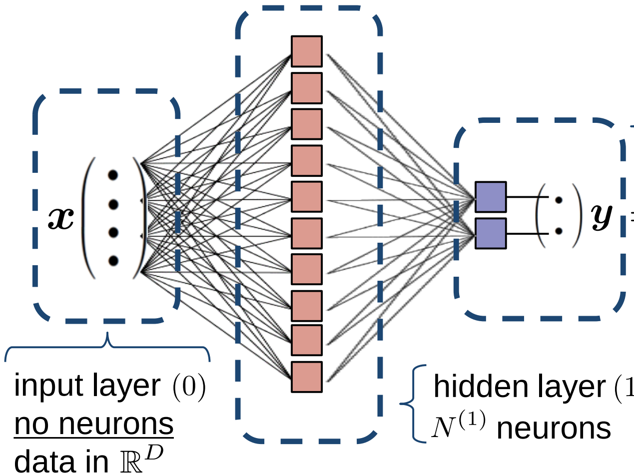
- Neurons can be **chained** together




$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**



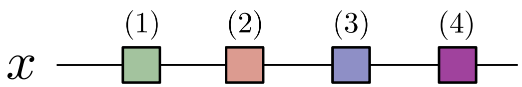

$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}^{(2)} + \mathbf{W}^{(2)} \cdot \sigma\left(\boldsymbol{b}^{(1)} + \mathbf{W}^{(1)} \cdot \boldsymbol{x}\right)\right)$$

$$\theta = \{\boldsymbol{b}^{(1)}, \mathbf{W}^{(1)}, \boldsymbol{b}^{(2)}, \mathbf{W}^{(2)}\} \in \mathbb{R}^{N^{(1)}} \times \mathbb{R}^{N^{(1)} \times D} \times \mathbb{R}^{N^{(2)}} \times \mathbb{R}^{N^{(2)} \times N^{(1)}}$$

$$\#\theta = N^{(1)} + N^{(1)}D + N^{(2)} + N^{(1)}N^{(2)}$$

# Artificial Neural Networks

- Neurons can be **chained** together

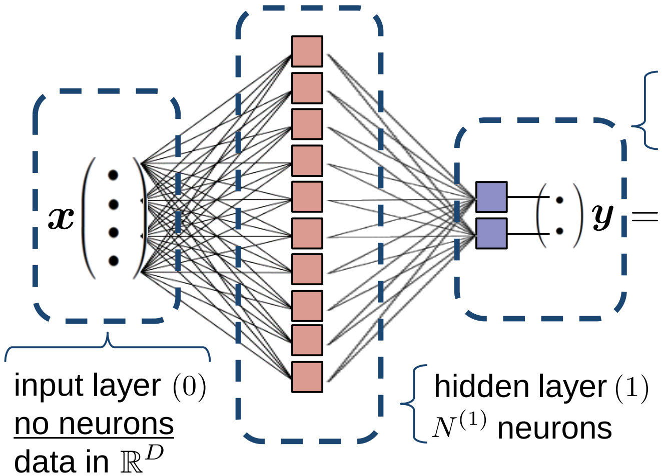



$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**

**Q:** *what happens if we choose $\sigma$ to be* ***linear*** *?*



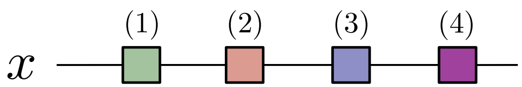

$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}^{(2)} + \mathbf{W}^{(2)} \cdot \sigma\left(\boldsymbol{b}^{(1)} + \mathbf{W}^{(1)} \cdot \boldsymbol{x}\right)\right)$$

$$\theta = \{\boldsymbol{b}^{(1)}, \mathbf{W}^{(1)}, \boldsymbol{b}^{(2)}, \mathbf{W}^{(2)}\} \in \mathbb{R}^{N^{(1)}} \times \mathbb{R}^{N^{(1)} \times D} \times \mathbb{R}^{N^{(2)}} \times \mathbb{R}^{N^{(2)} \times N^{(1)}}$$

$$\#\theta = N^{(1)} + N^{(1)}D + N^{(2)} + N^{(1)}N^{(2)}$$

# Artificial Neural Networks

- Neurons can be **chained** together

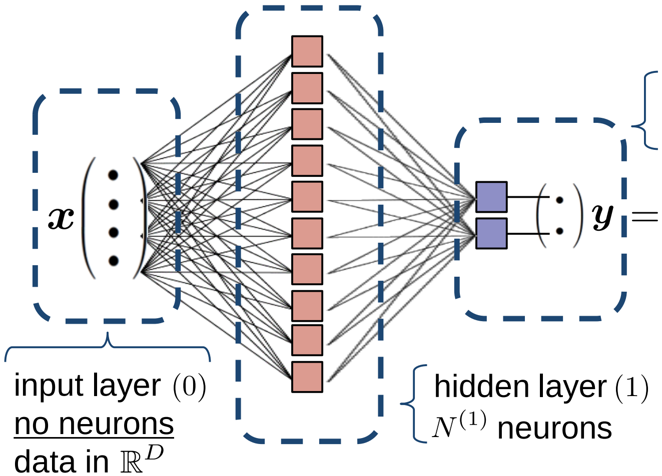



$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**

**Q:** *what happens if we choose $\sigma$ to be* ***linear*** *?*
- ▸ *$f_\theta$ is then linear*



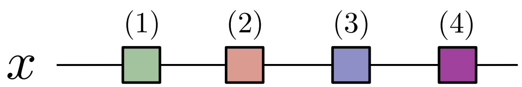

$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}^{(2)} + \mathbf{W}^{(2)} \cdot \sigma\left(\boldsymbol{b}^{(1)} + \mathbf{W}^{(1)} \cdot \boldsymbol{x}\right)\right)$$

$$\theta = \{\boldsymbol{b}^{(1)}, \mathbf{W}^{(1)}, \boldsymbol{b}^{(2)}, \mathbf{W}^{(2)}\} \in \mathbb{R}^{N^{(1)}} \times \mathbb{R}^{N^{(1)} \times D} \times \mathbb{R}^{N^{(2)}} \times \mathbb{R}^{N^{(2)} \times N^{(1)}}$$

$$\#\theta = N^{(1)} + N^{(1)}D + N^{(2)} + N^{(1)}N^{(2)}$$

# Artificial Neural Networks

- Neurons can be **chained** together

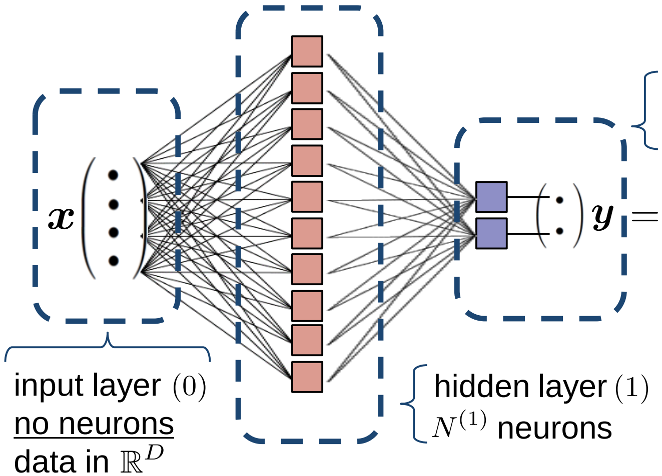



$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**

**Q:** *what happens if we choose $\sigma$ to be **linear** ?*
- ▸ *$f_\theta$ is then linear*




$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}^{(2)} + \mathbf{W}^{(2)} \cdot \sigma\left(\boldsymbol{b}^{(1)} + \mathbf{W}^{(1)} \cdot \boldsymbol{x}\right)\right)$$

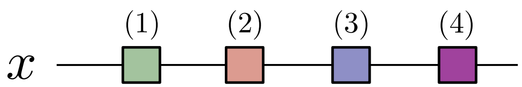
$$\theta = \{\boldsymbol{b}^{(1)}, \mathbf{W}^{(1)}, \boldsymbol{b}^{(2)}, \mathbf{W}^{(2)}\} \in \mathbb{R}^{N^{(1)}} \times \mathbb{R}^{N^{(1)} \times D} \times \mathbb{R}^{N^{(2)}} \times \mathbb{R}^{N^{(2)} \times N^{(1)}}$$

$$\#\theta = N^{(1)} + N^{(1)}D + N^{(2)} + N^{(1)}N^{(2)}$$

**Universal Approximation Theorem** [Hornik, Stinchcomb, White, 1991]: *"A **single hidden layer** neural network with any **"sigmoid-like"** activation function and with a **linear output** unit can approximate any continuous function arbitrarily well, for **sufficiently large** $N^{(1)}$."*

# Artificial Neural Networks

- Neurons can be **chained** together



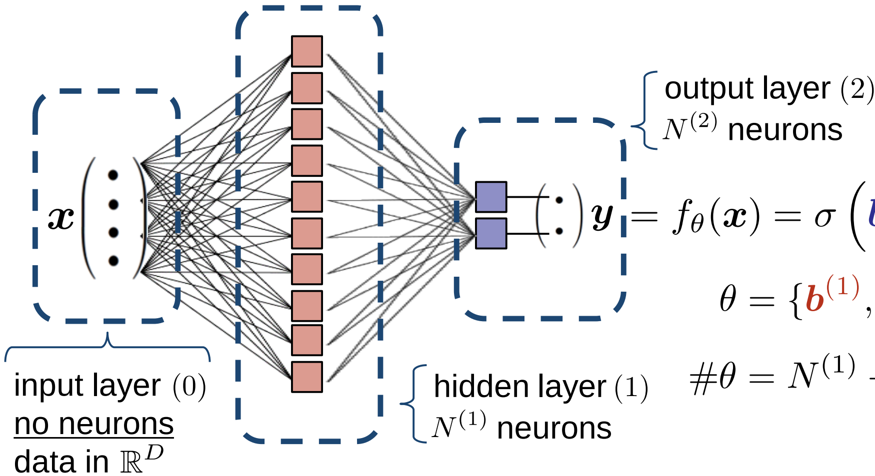

$$y = f_\theta(x) = \sigma\left(b^{(4)} + w^{(4)} \cdot \sigma\left(b^{(3)} + w^{(3)} \cdot \sigma\left(b^{(2)} + w^{(2)} \cdot \sigma\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right)$$

$$\theta = \{b^{(1)}, w^{(1)}, b^{(2)}, w^{(2)}, b^{(3)}, w^{(3)}, b^{(4)}, w^{(4)}\} \in \mathbb{R}^8$$

- Perceptron with a **single hidden layer**

**Q:** *what happens if we choose $\sigma$ to be* ***linear*** *?*

- ▸ *$f_\theta$ is then linear*




$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = \sigma\left(\boldsymbol{b}^{(2)} + \mathbf{W}^{(2)} \cdot \sigma\left(\boldsymbol{b}^{(1)} + \mathbf{W}^{(1)} \cdot \boldsymbol{x}\right)\right)$$

$$\theta = \{\boldsymbol{b}^{(1)}, \mathbf{W}^{(1)}, \boldsymbol{b}^{(2)}, \mathbf{W}^{(2)}\} \in \mathbb{R}^{N^{(1)}} \times \mathbb{R}^{N^{(1)} \times D} \times \mathbb{R}^{N^{(2)}} \times \mathbb{R}^{N^{(2)} \times N^{(1)}}$$

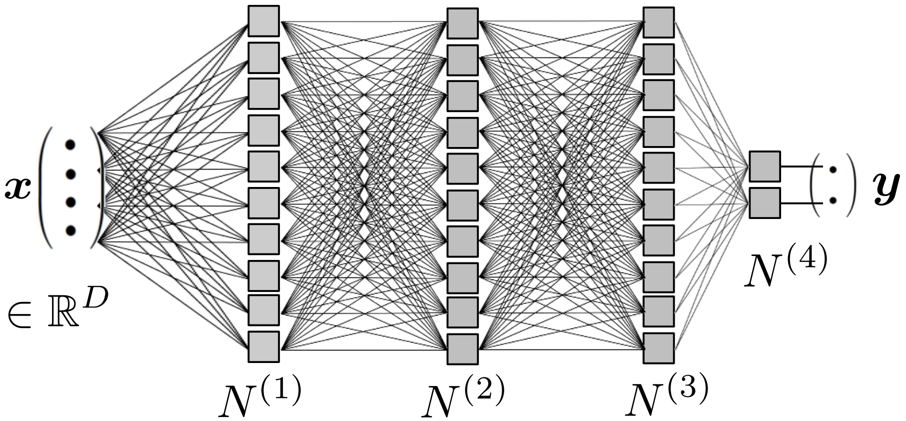
$$\#\theta = N^{(1)} + N^{(1)}D + N^{(2)} + N^{(1)}N^{(2)}$$

**Universal Approximation Theorem** [Hornik, Stinchcomb, White, 1991]: *“A* ***single hidden layer*** *neural network with any* ***“sigmoid-like”*** *activation function and with a* ***linear output*** *unit can approximate any continuous function arbitrarily well, for* ***sufficiently large*** *$N^{(1)}$.”*

… But the required $N^{(1)}$ may be **exponential** in the input dimension $D$ .

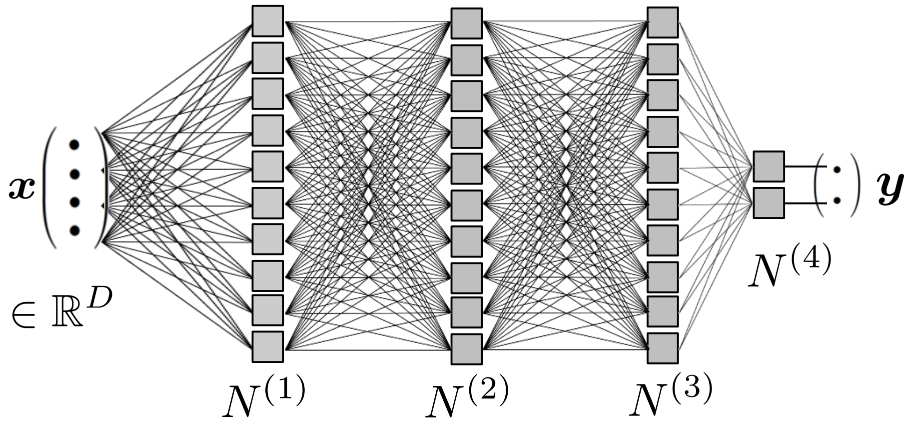
# Artificial Neural Networks

- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)



# Artificial Neural Networks

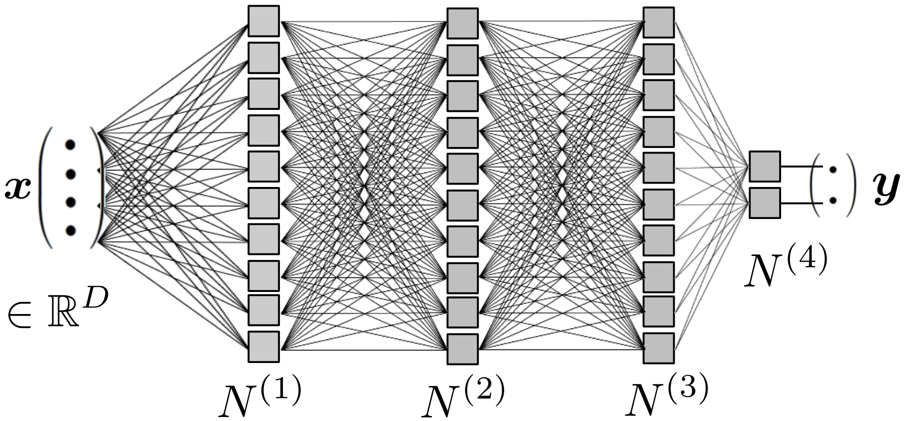
- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)




***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$*,*
*how many parameters?*

# Artificial Neural Networks

- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)

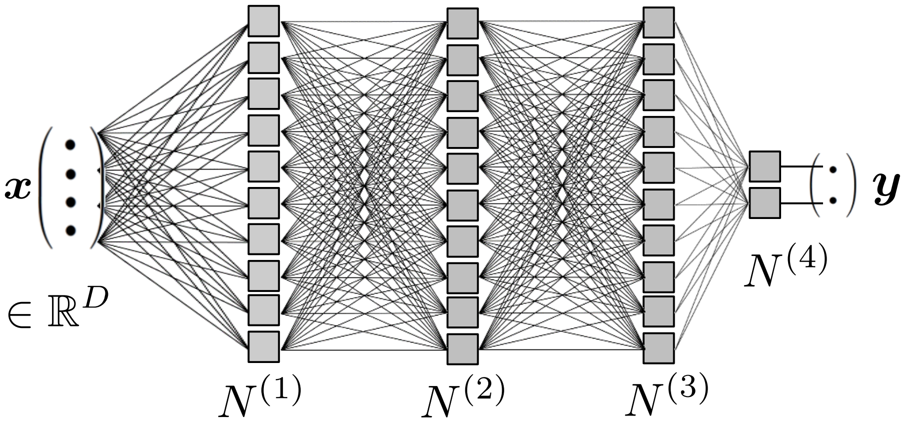



***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$*,*
*how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$

# Artificial Neural Networks

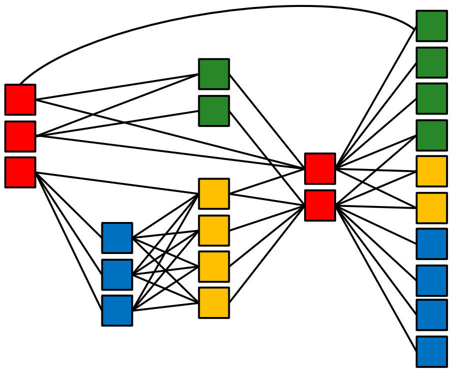
- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)

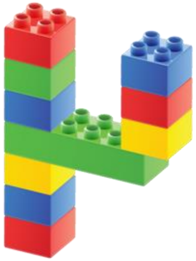



***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$, *how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$

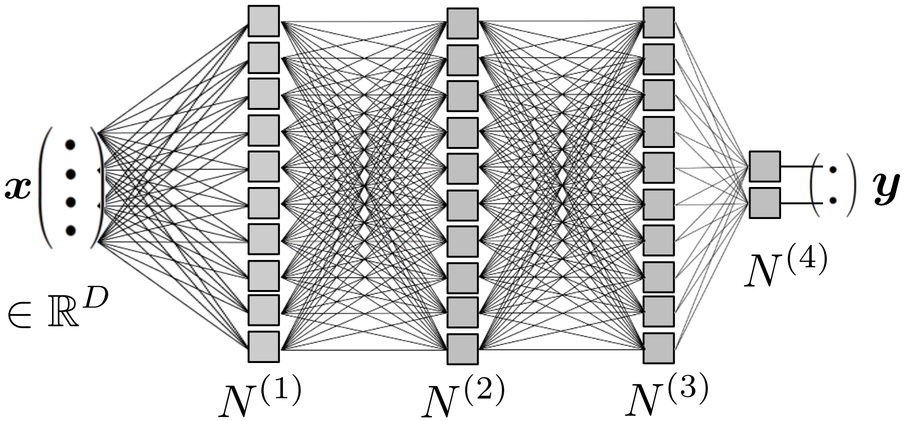
•Artificial neurons, as **elementary computing units**, can be combined in many different ways





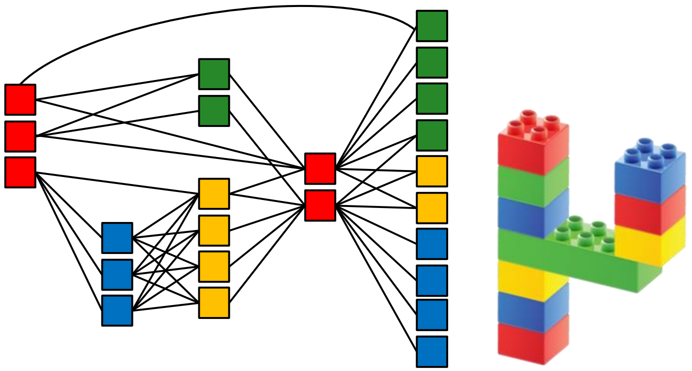
# Artificial Neural Networks

- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)




***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$, *how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$

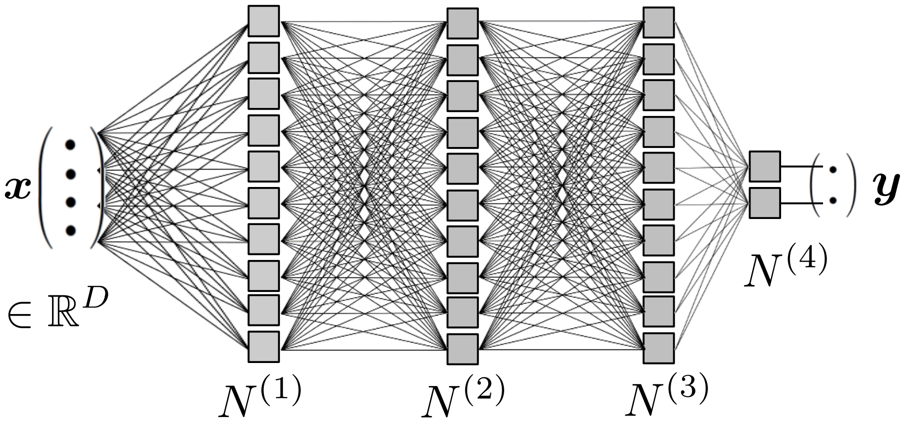


•Artificial neurons, as **elementary computing units**, can be combined in many different ways

•No cycle in the graph = **Feedforward Neural Networks**

# Artificial Neural Networks

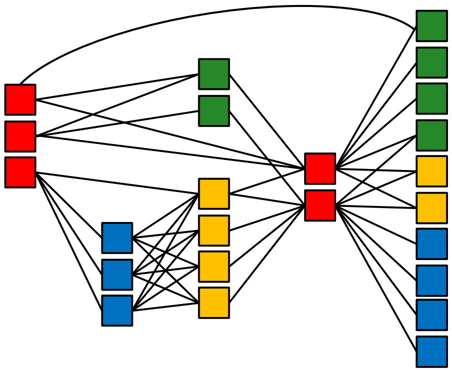
- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)

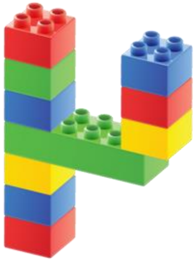



***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$*,*
*how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$

- Artificial neurons, as **elementary computing units**, can be combined in many different ways

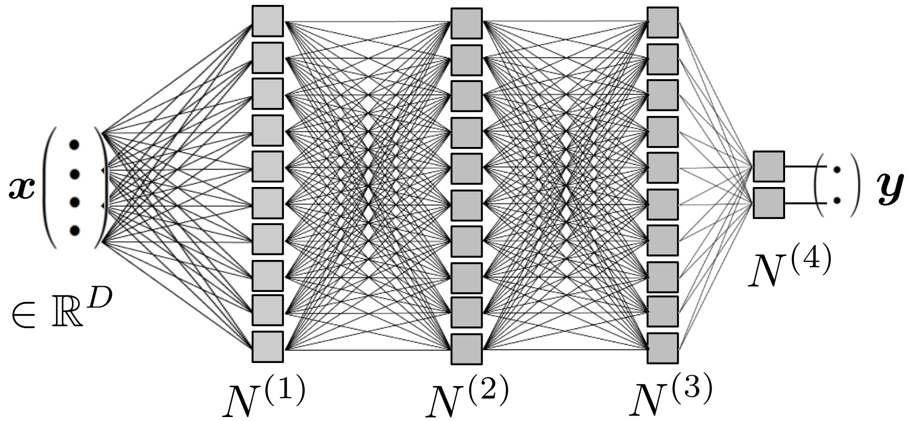




- No cycle in the graph = **Feedforward Neural Networks**
- We call a given network of neurons an **architecture**

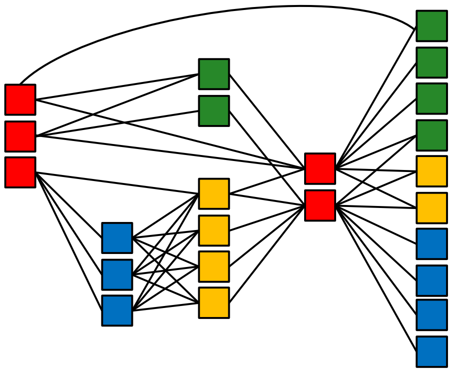
# Artificial Neural Networks

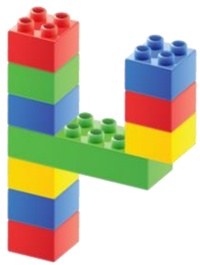
- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)




***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$*,*
*how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$





•Artificial neurons, as **elementary computing units**,
can be combined in many different ways

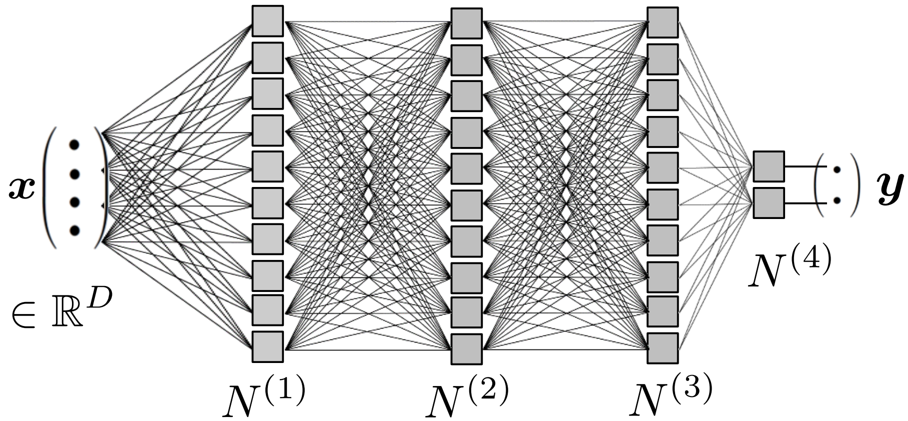
•No cycle in the graph = **Feedforward Neural Networks**

•We call a given network of neurons an **architecture**

•Neural Networks form a very flexible **family** of **parameterized families** of **non-linear functions:** $\{\mathcal{F}_i\}_i$ where $\mathcal{F}_i = \{f^i_\theta\}_\theta$

# Artificial Neural Networks

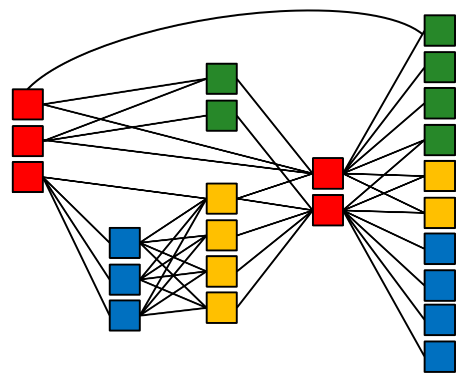
- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)

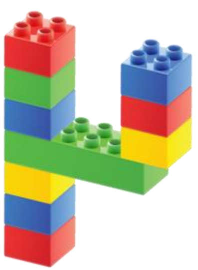



***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$*,*
*how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$

•Artificial neurons, as **elementary computing units**,
can be combined in many different ways





•No cycle in the graph = **Feedforward Neural Networks**

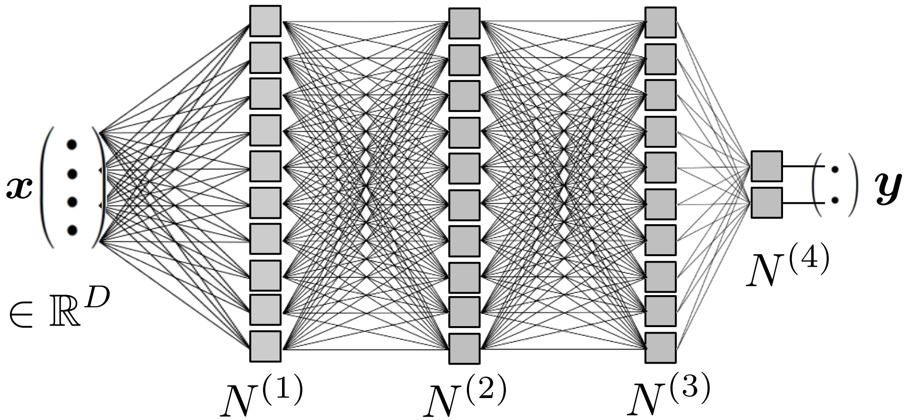
•We call a given network of neurons an **architecture**

•Neural Networks form a very flexible **family** of **parameterized families** of **non-linear functions:** $\{\mathcal{F}_i\}_i$ where $\mathcal{F}_i = \{f_\theta^i\}_\theta$

Artificial neural networks with 2 or more hidden layers are called **Deep Neural Networks** (DNNs)

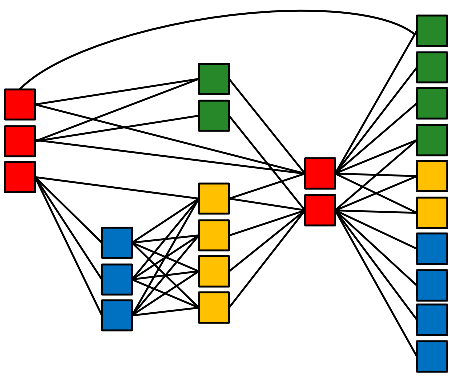
# Artificial Neural Networks

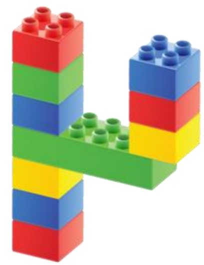
- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)




***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$*,*
*how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$





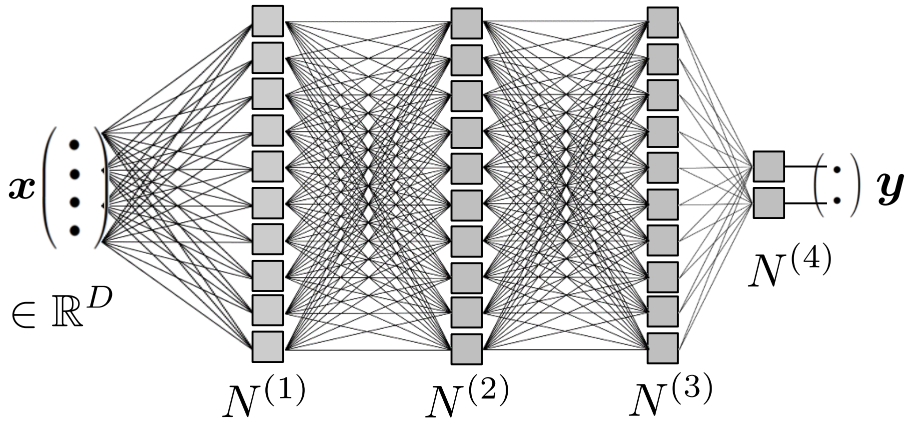
- Artificial neurons, as **elementary computing units**, can be combined in many different ways
- No cycle in the graph = **Feedforward Neural Networks**
- We call a given network of neurons an **architecture**
- Neural Networks form a very flexible **family** of **parameterized families** of **non-linear functions:** $\{\mathcal{F}_i\}_i$ where $\mathcal{F}_i = \{f_\theta^i\}_\theta$

Artificial neural networks with 2 or more hidden layers are called **Deep Neural Networks** (DNNs)

*How to **fit** a **DNN model**?*

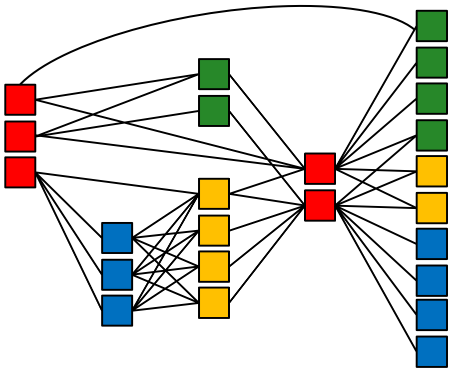
# Artificial Neural Networks

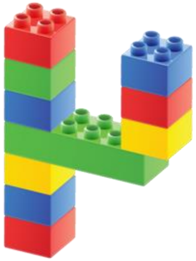
- **Multilayer Perceptron** (MLP) with 3 hidden layer (***Depth 4***)




***Exercice:***
*Suppose* $D = N^{(1)} = N^{(2)} = N^{(3)} = N^{(4)} = N$*,*
*how many parameters?*

$$\#\theta = 4(N + N^2) = \mathcal{O}(N^2)$$





•Artificial neurons, as **elementary computing units**, can be combined in many different ways

•No cycle in the graph = **Feedforward Neural Networks**

•We call a given network of neurons an **architecture**

•Neural Networks form a very flexible **family** of **parameterized families** of **non-linear functions:** $\{\mathcal{F}_i\}_i$ where $\mathcal{F}_i = \{f_\theta^i\}_\theta$

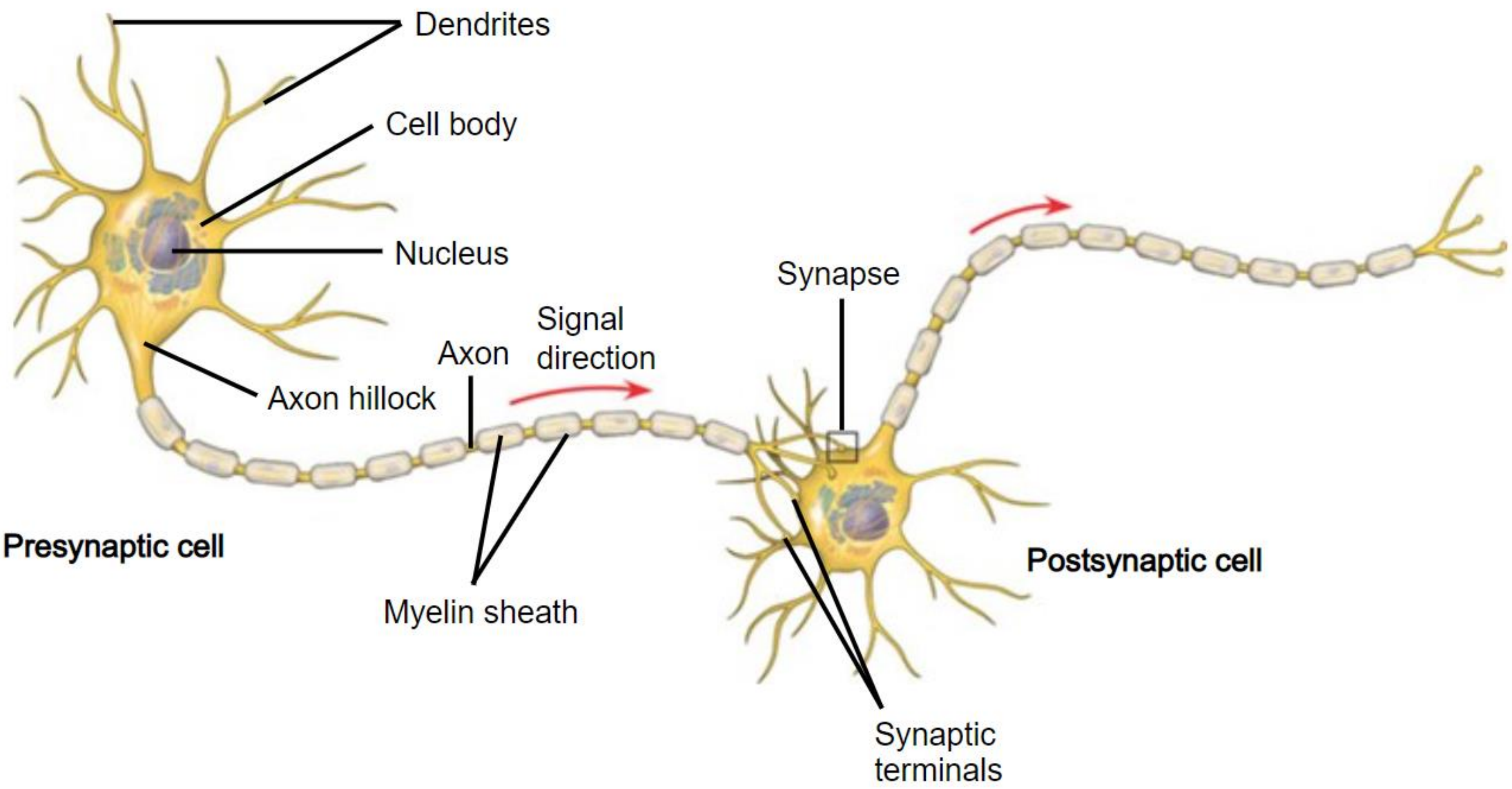
Artificial neural networks with 2 or more hidden layers are called **Deep Neural Networks** (DNNs)

*How to **fit** a **DNN model**?*

⇨ Next chapter :-)

# Why Deep?

1) Biological inspiration



# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| | |
| | |
| | |
| | |
| | |
| | |

# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| | |
| | |
| | |
| | |
| | |

# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
| --- | --- |
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| Asynchronous | Synchronous |
| | |
| | |
| | |
| | |

# Why Deep?

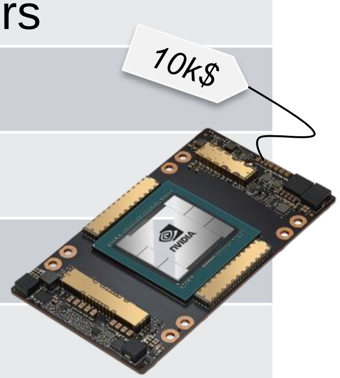
1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| Asynchronous | Synchronous |
| Very plastic architecture | (Mostly) fixed architecture |
| | |
| | |
| | |

# Why Deep?

1) Biological inspiration

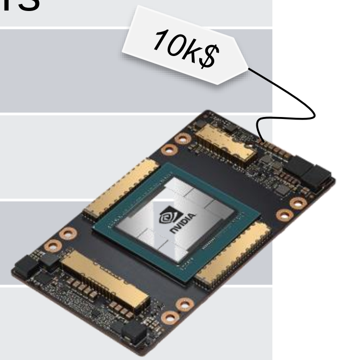
| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| Asynchronous | Synchronous |
| Very plastic architecture | (Mostly) fixed architecture |
| 12 W of power | Nvidia A100 GPU : 300 W* |
| | |
| | |



# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| Asynchronous | Synchronous |
| Very plastic architecture | (Mostly) fixed architecture |
| 12 W of power | Nvidia A100 GPU : 300 W* |
| | |
| | |

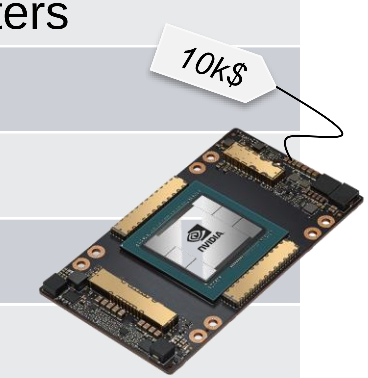



* - A chat session with chatGPT mobilizes roughly a full A100. Training GPT-3 is estimated to have taken ~81 years of A100.
  - GPT-3 has ~135 billion parameters and roughly ~0.25 billion “neurons”.

# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| Asynchronous | Synchronous |
| Very plastic architecture | (Mostly) fixed architecture |
| 12 W of power | Nvidia A100 GPU : 300 W* |
| Biological neuron = extremely complex** and not fully understood | Artificial Neuron = a weighted sum and a threshold. |
| | |

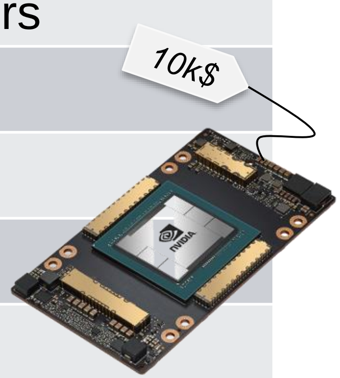



* - A chat session with chatGPT mobilizes roughly a full A100. Training GPT-3 is estimated to have taken ~81 years of A100.
- GPT-3 has ~135 billion parameters and roughly ~0.25 billion "neurons".

# Why Deep?

1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| Asynchronous | Synchronous |
| Very plastic architecture | (Mostly) fixed architecture |
| 12 W of power | Nvidia A100 GPU : 300 W* |
| Biological neuron = extremely complex** and not fully understood | Artificial Neuron = a weighted sum and a threshold. |
| | |



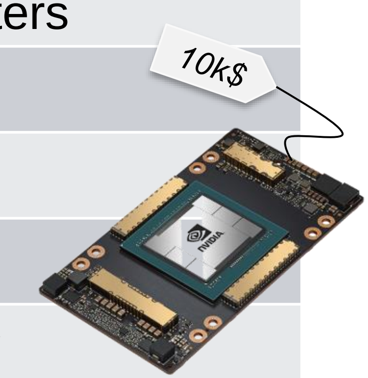

* - A chat session with chatGPT mobilizes roughly a full A100. Training GPT-3 is estimated to have taken ~81 years of A100.
 - GPT-3 has ~135 billion parameters and roughly ~0.25 billion "neurons".

** [Beniagev et al. 2021]: a single bio-neuron is well approximated by a 1000-neuron ANN of depth 5-8

# Why Deep?

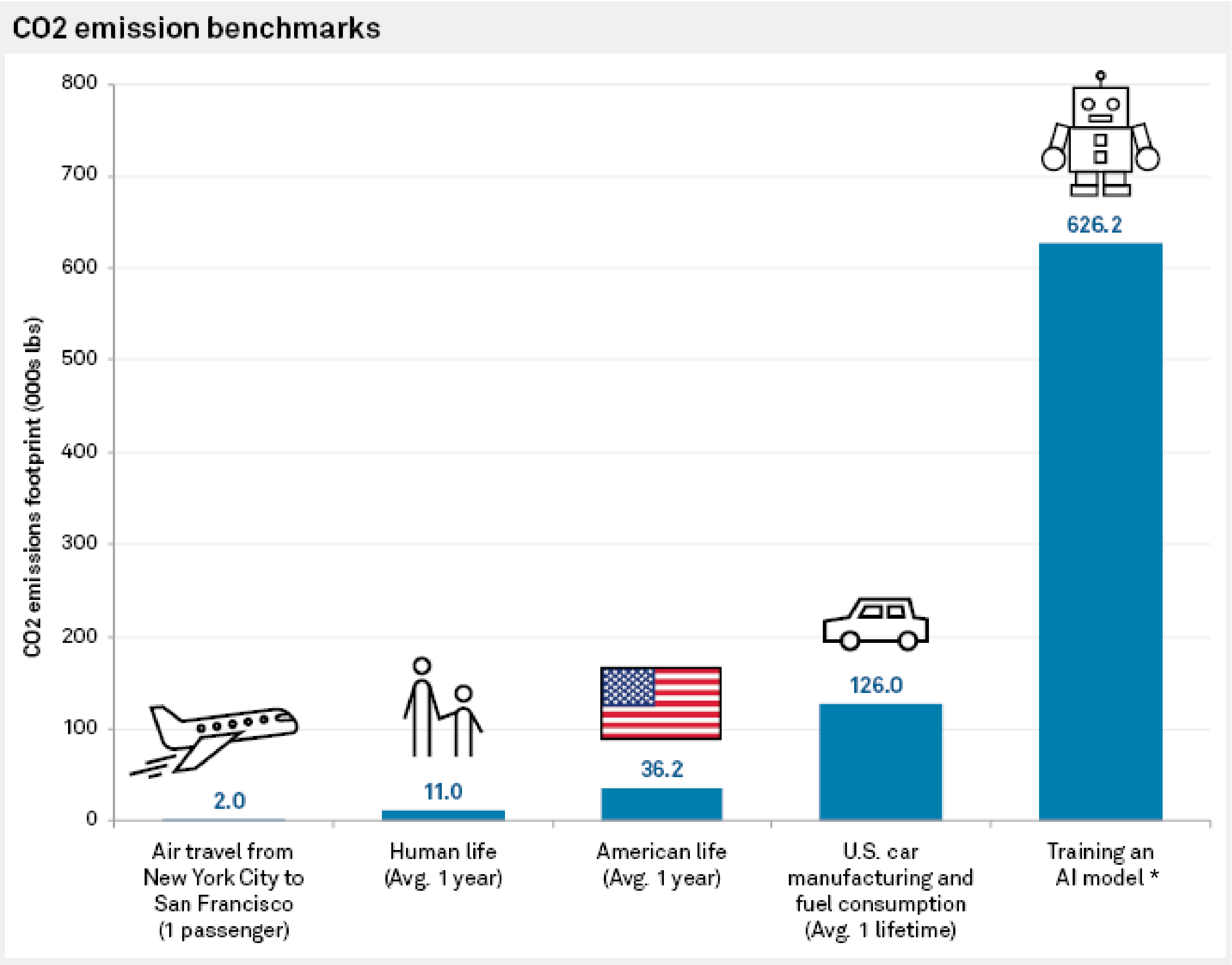
1) Biological inspiration

| Human Brain | Artificial Neural Network |
|---|---|
| ~ 86 billion neurons | 100k - 1 billion neurons |
| ~ 7,000 synapse connections per neuron (~600 trillion connections) | 3 - 1,000 connections per neuron<br>1 million - 1 trillion parameters |
| Massively parallel | (Mostly) sequential |
| Asynchronous | Synchronous |
| Very plastic architecture | (Mostly) fixed architecture |
| 12 W of power | Nvidia A100 GPU : 300 W* |
| Biological neuron = extremely complex** and not fully understood | Artificial Neuron = a weighted sum and a threshold. |
| Inherited from 3.7B year of evolution | Programmed & designed by an engineer |




* - A chat session with chatGPT mobilizes roughly a full A100. Training GPT-3 is estimated to have taken ~81 years of A100.
  - GPT-3 has ~135 billion parameters and roughly ~0.25 billion “neurons”.

** [Beniagev et al. 2021]: a single bio-neuron is well approximated by a 1000-neuron ANN of depth 5-8




[Strubell et Al. (2019)], chart by Oslo University

* 213M parameters NLP Transformer with neural architecture search.

Training <u>one</u> instance of GPT-3:
≈ **167** klbs $CO_2e$

Operating chatGPT in Feb. 2023:
≈ **2046** klbs $CO_2e$

# Why Deep?

2) Bypassing feature engineering

# Why Deep?

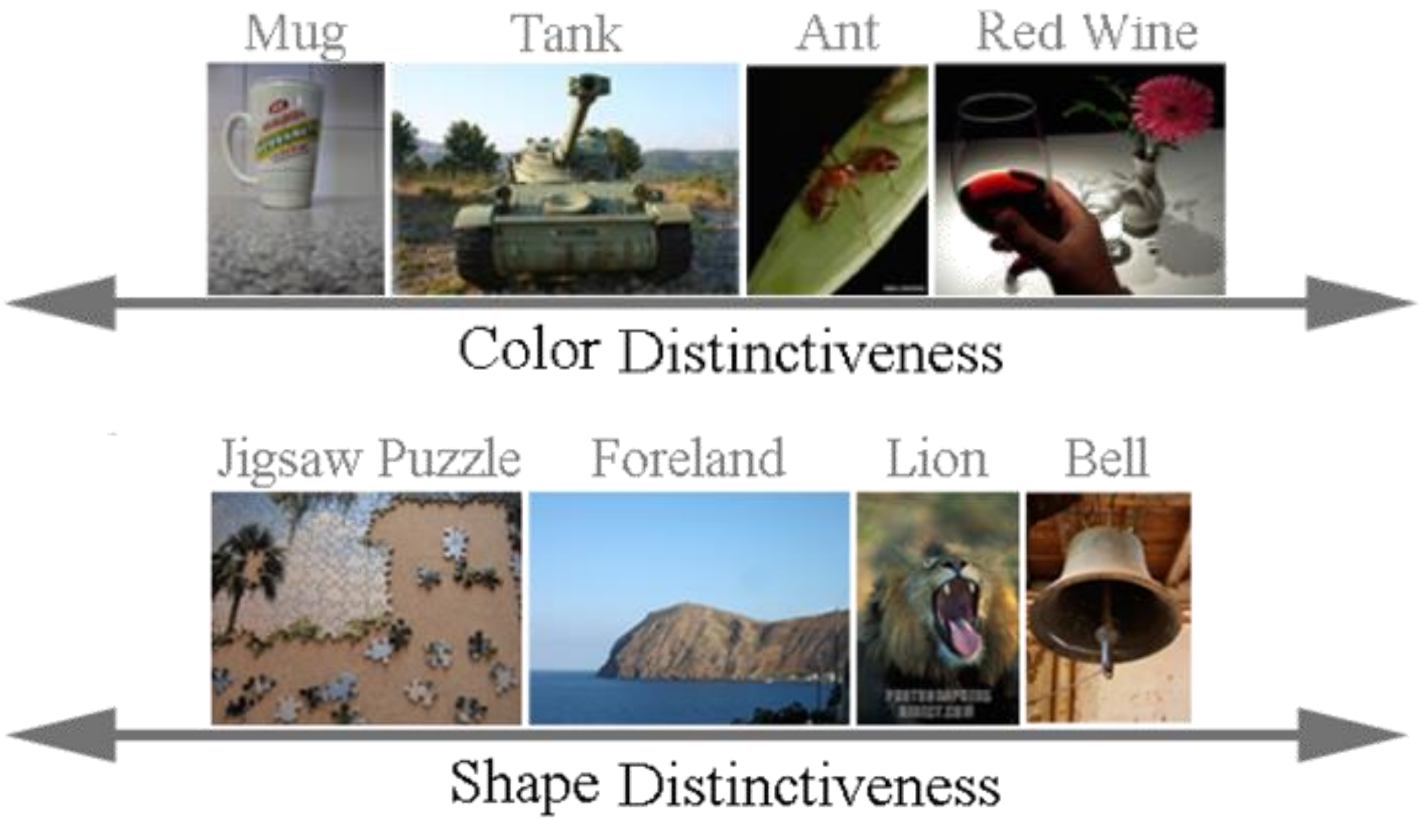
2) Bypassing feature engineering

- For many tasks, **manually defining** relevant features is hard

# Why Deep?

2) Bypassing feature engineering

- For many tasks, **manually defining** relevant features is hard
- Examples of data variability from the ImageNet dataset:



# Why Deep?

2) Bypassing feature engineering

- For many tasks, **manually defining** relevant features is hard
- Examples of data variability from the ImageNet dataset:

# Why Deep?

## 2) Bypassing feature engineering

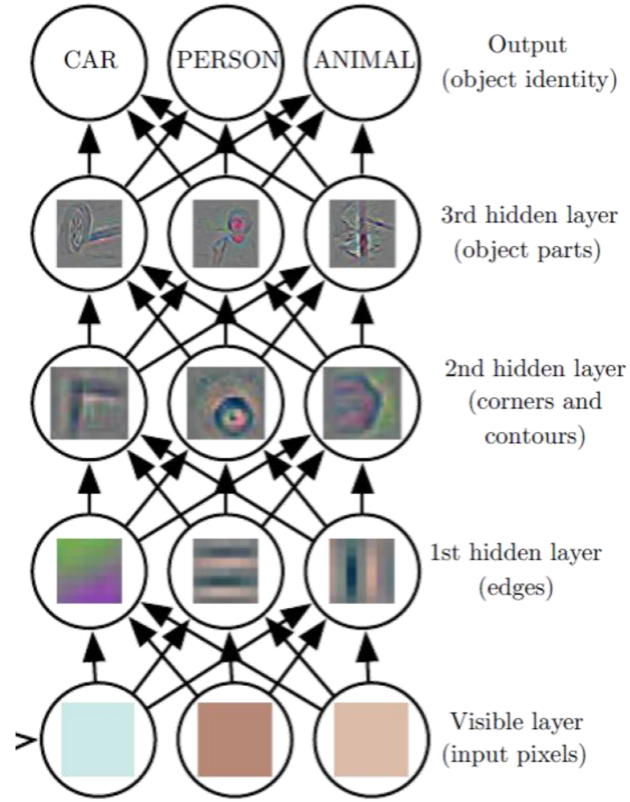
- The early successes of deep learning (1998, 2012) were in **image classification**, because they proved to be very efficient at **representation learning.**

# Why Deep?

## 2) Bypassing feature engineering

- The early successes of deep learning (1998, 2012) were in **image classification**, because they proved to be very efficient at **representation learning.**

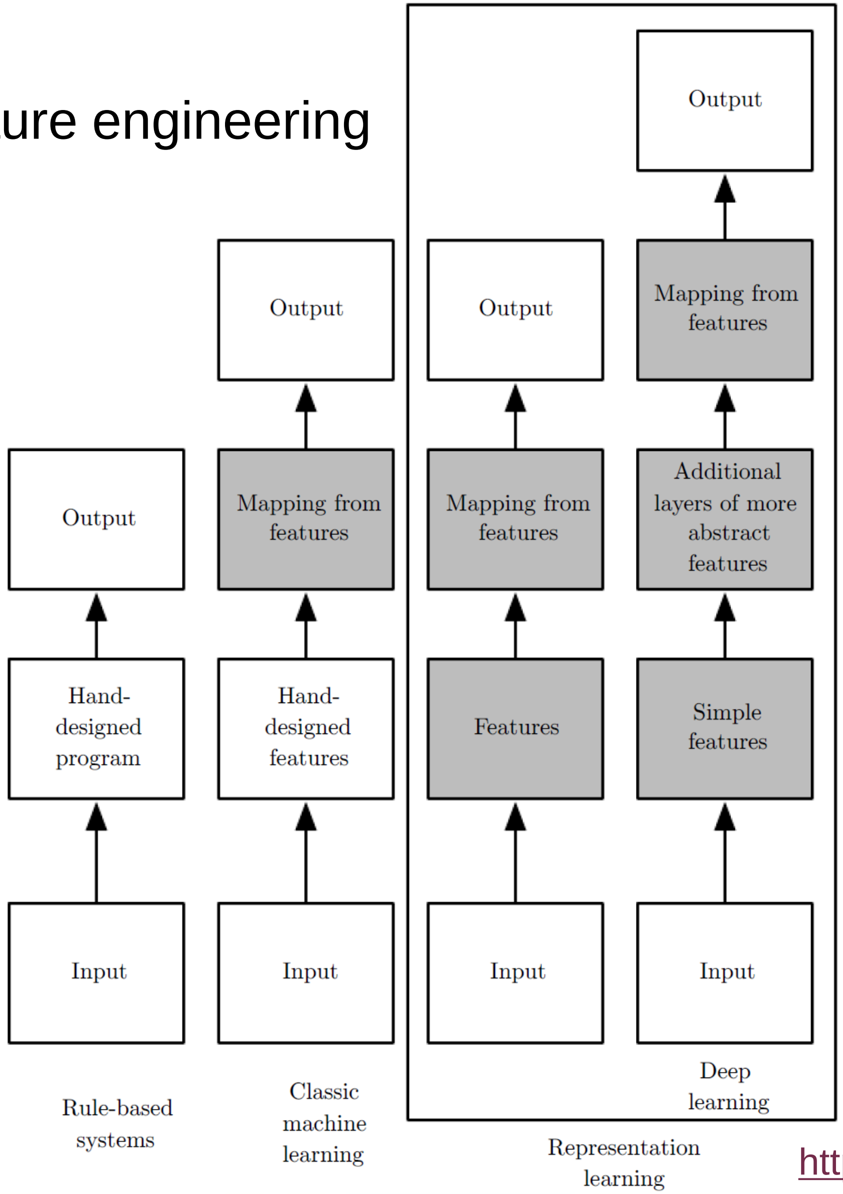
- Starting from **raw pixel values** in color channels, the layers of a **deep *convolutional* network**, seem to learn **more and more elaborate features** as the depth increase




*(more explanations on this figure later!)*

# Why Deep?

2) Bypassing feature engineering




https://www.deeplearningbook.org/

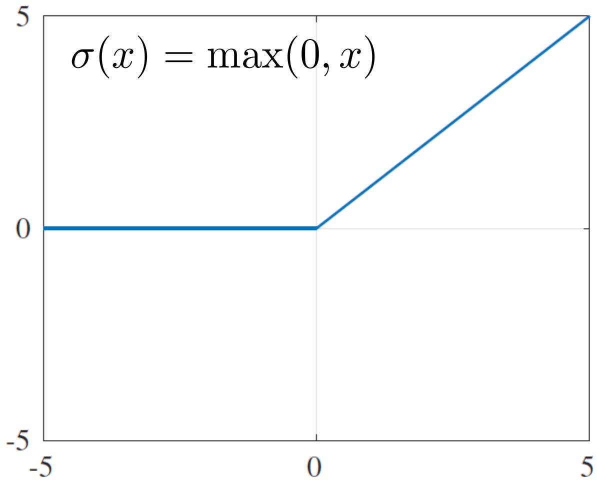
# Why Deep?

3) The “Origami Effect”

# Why Deep?

## 3) The “Origami Effect”

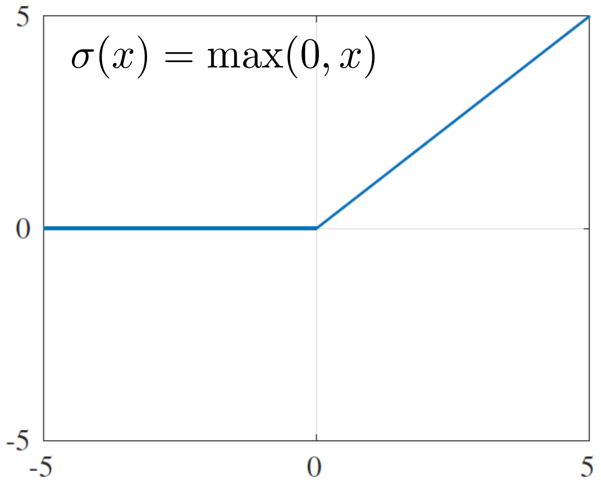
- ReLU activation:



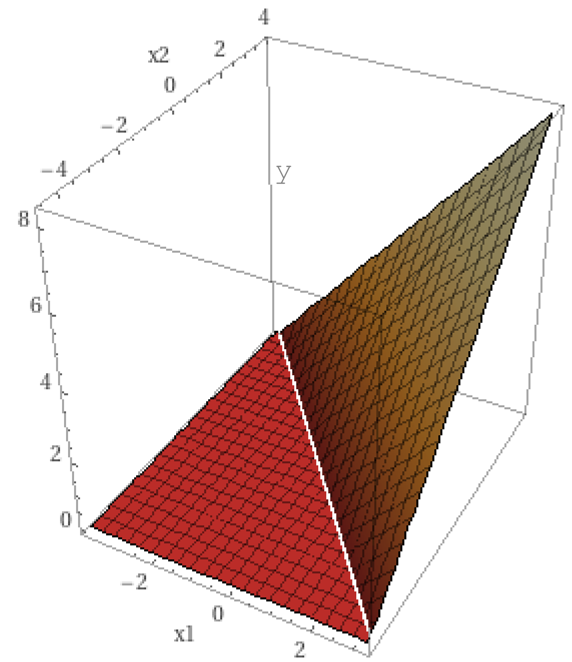
# Why Deep?

## 3) The "Origami Effect"

- ReLU activation:



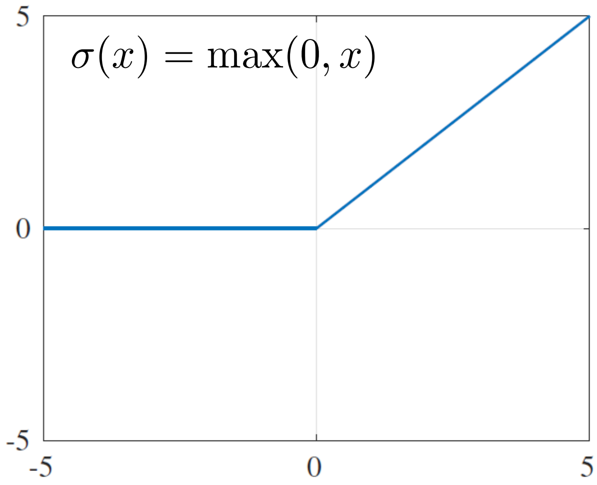

- For a 2D input:
  $y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$



# Why Deep?

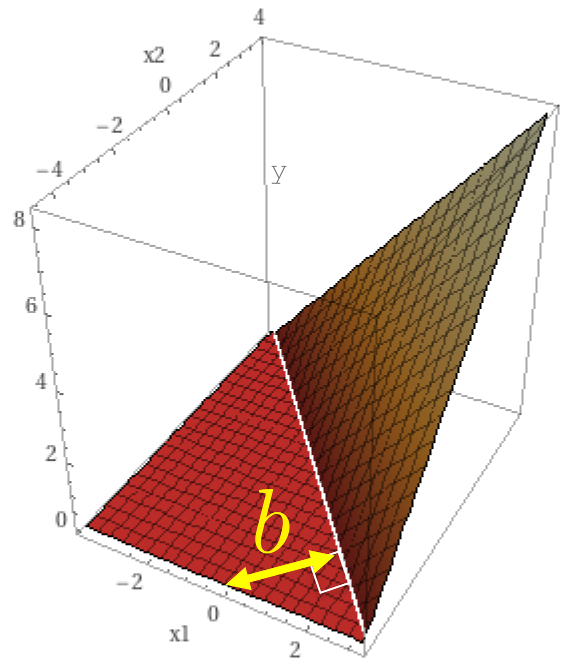
## 3) The "Origami Effect"

- ReLU activation:




- For a 2D input:

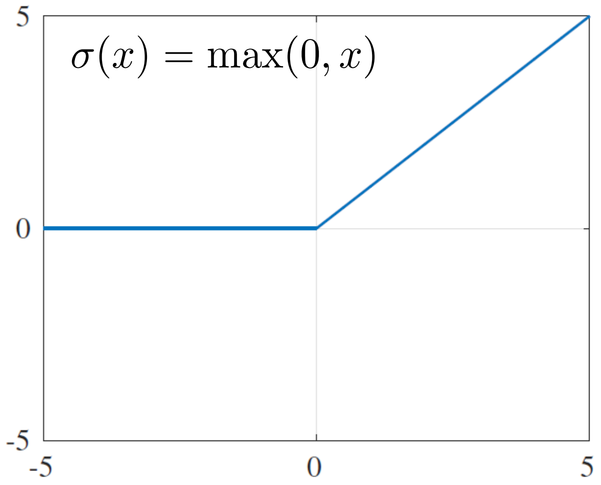
$$y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$$



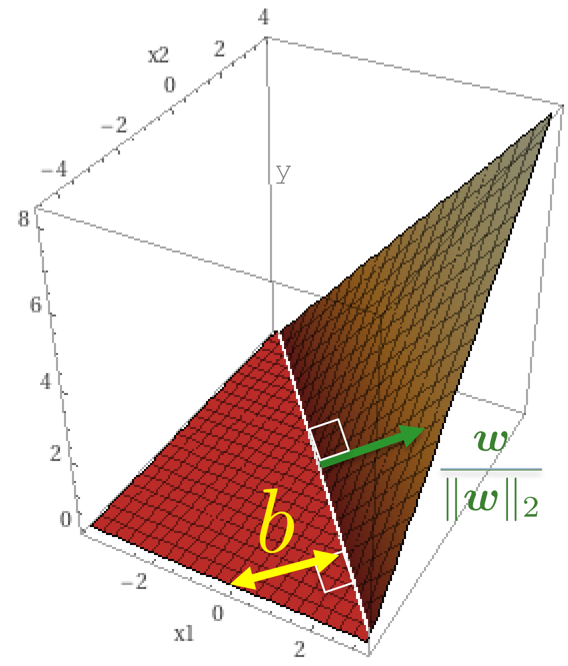
# Why Deep?

3) The "Origami Effect"

- ReLU activation:



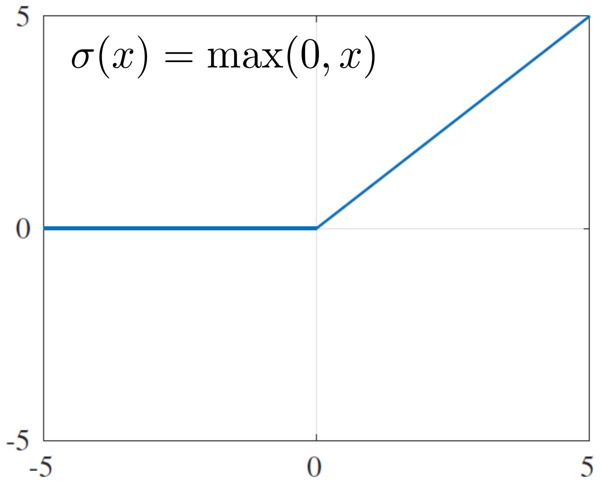

- For a 2D input:
  $y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$



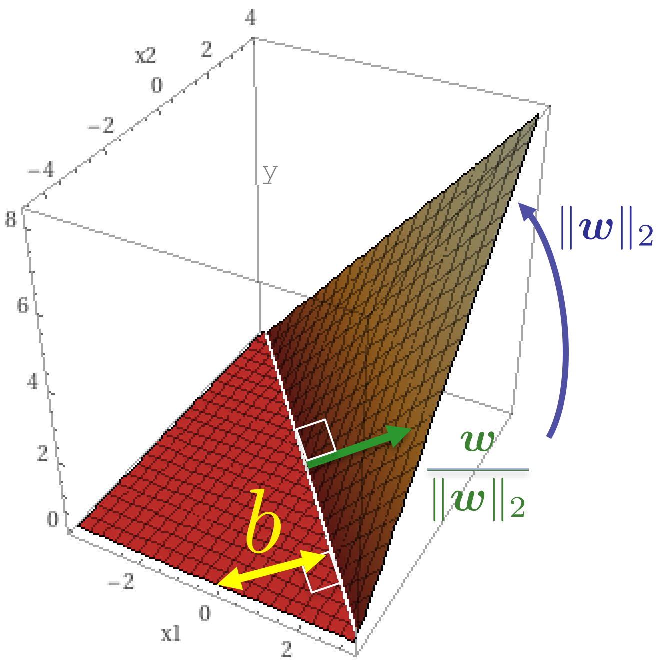
# Why Deep?

3) The "Origami Effect"

- ReLU activation:



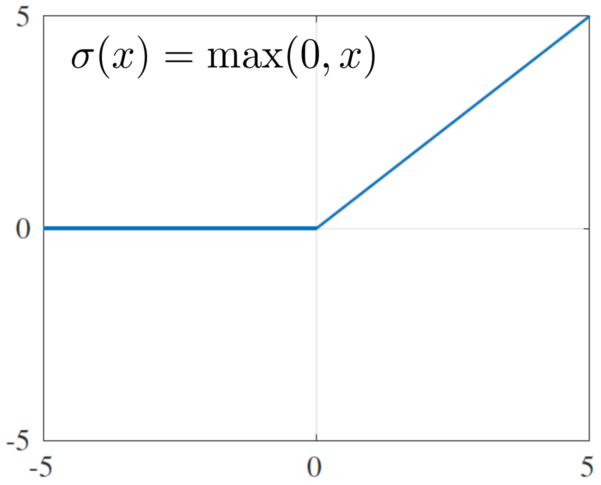

- For a 2D input:
  $y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$



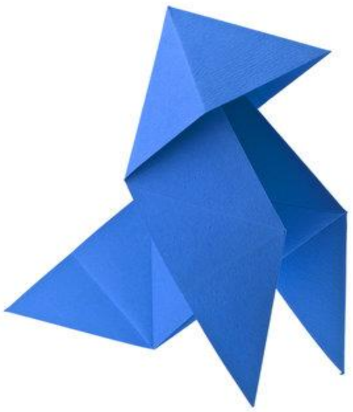
# Why Deep?

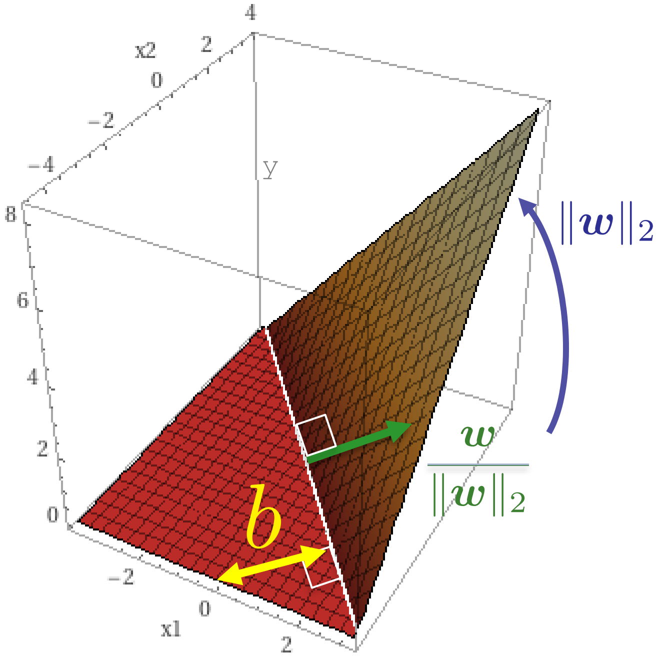
## 3) The “Origami Effect”

- ReLU activation:




- For a 2D input:
  $y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$

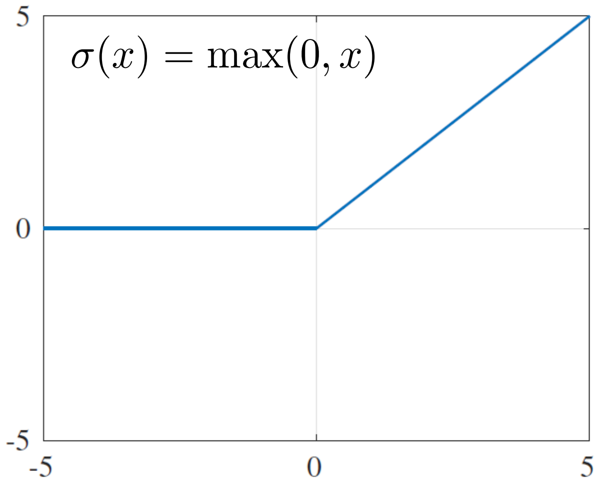




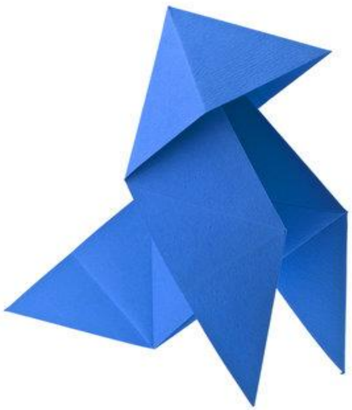
# Why Deep?

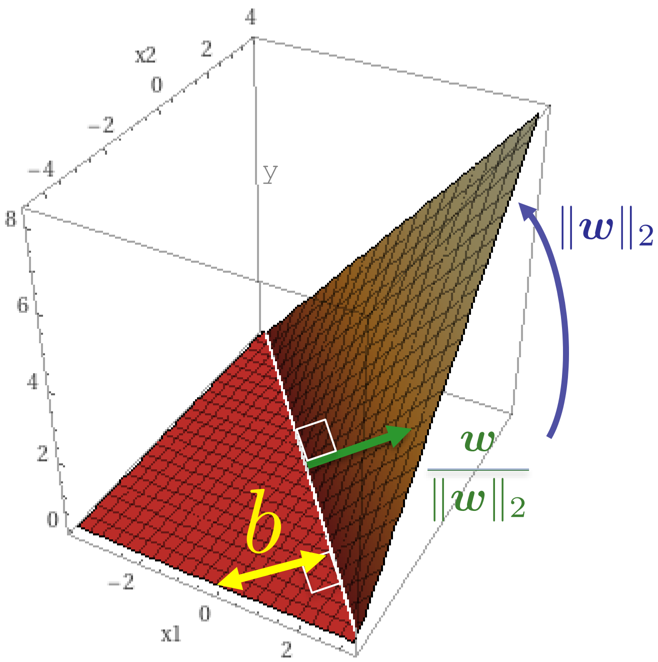
3) The "Origami Effect"

- ReLU activation:




- For a 2D input:
  $y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$





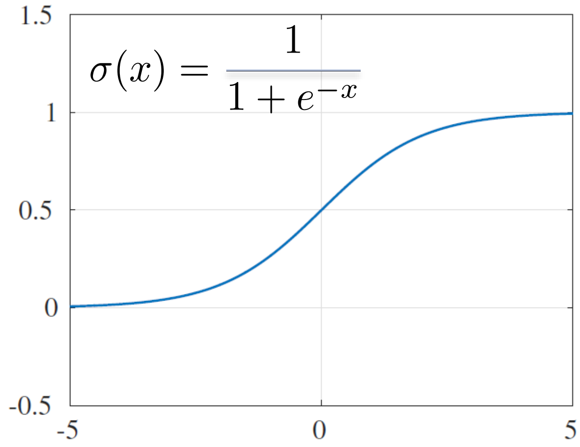

**Theorem** [G. Montufar et al., 2014]: *the max. number of linear regions modeled by a piecewise linear network (i.e., a network with ReLU neurons) with D inputs, L layers, and N units per layer is in the order of*

$$N^D \left(\frac{N}{D}\right)^{D(L-2)}$$

*, i.e., the model capacity is **exponential in the depth** L, i.e., in the **model size** (recall it is O(LN²) )*

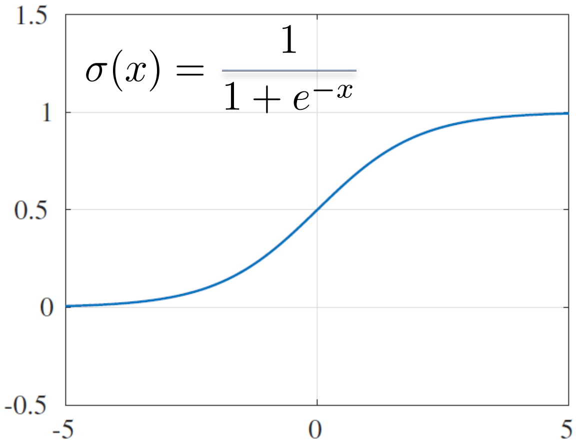
# Why Deep?

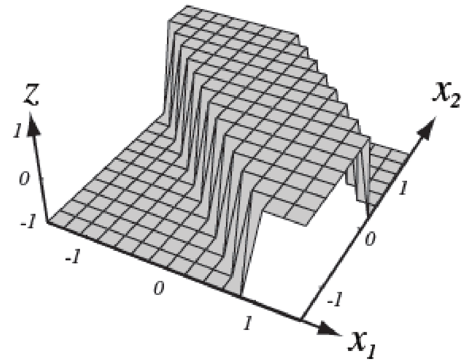
## 3) The “Origami Effect”: Illustration with Sigmoids

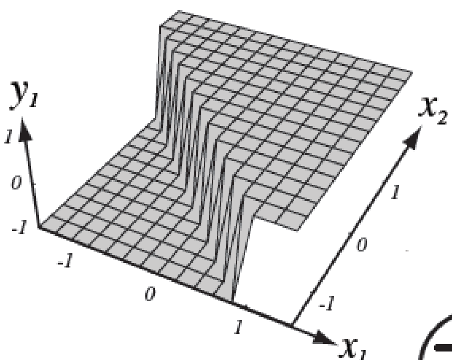


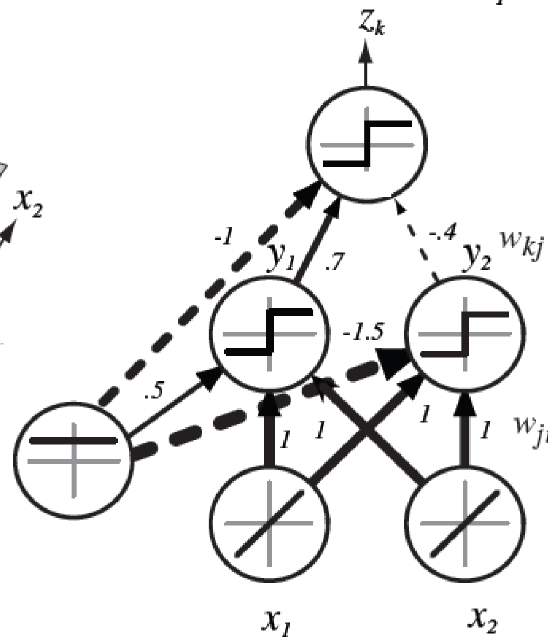
# Why Deep?

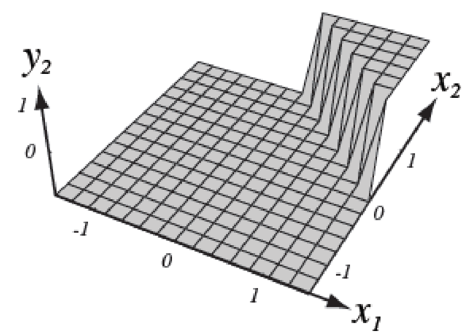
## 3) The “Origami Effect”: Illustration with Sigmoids



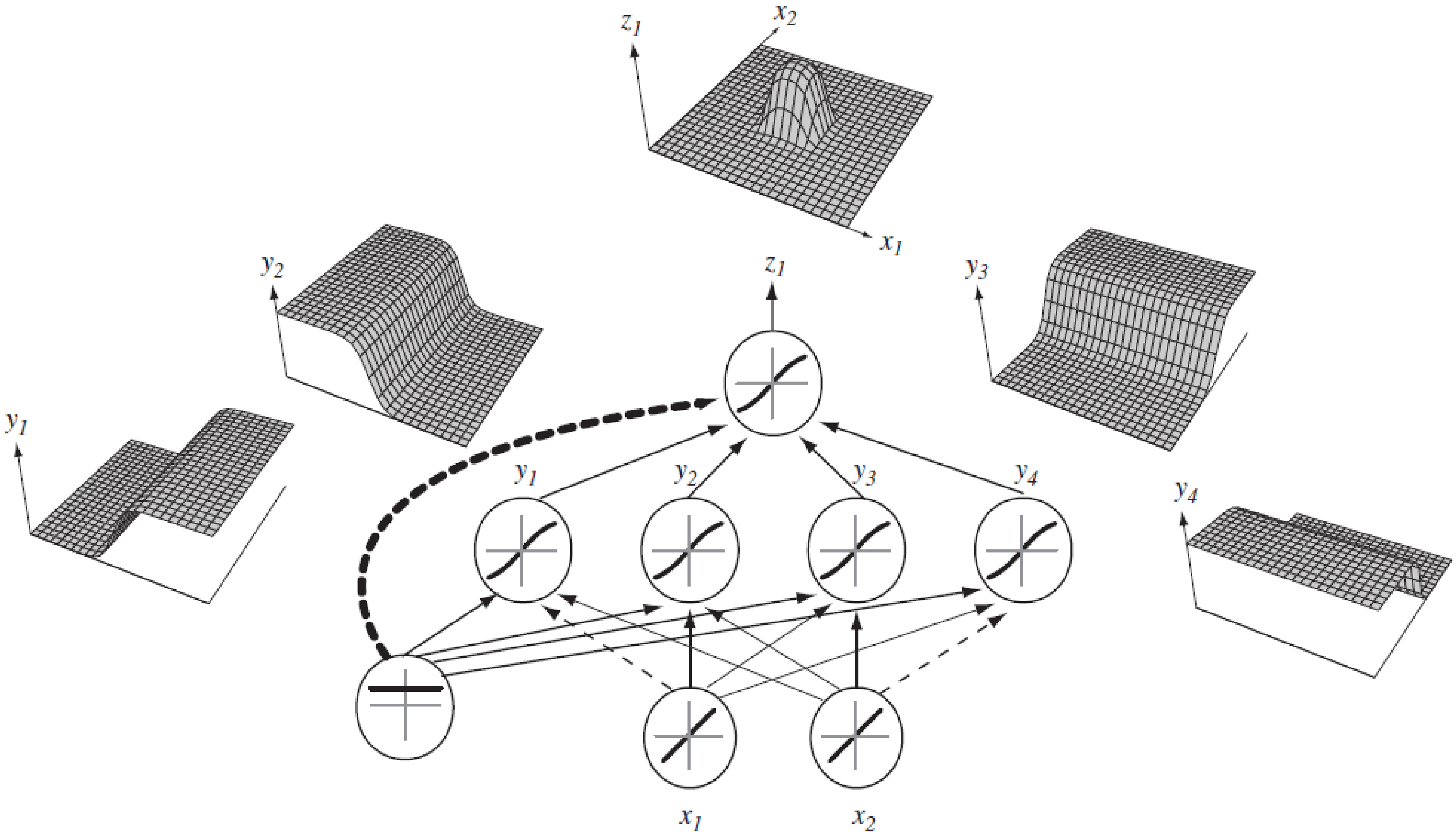



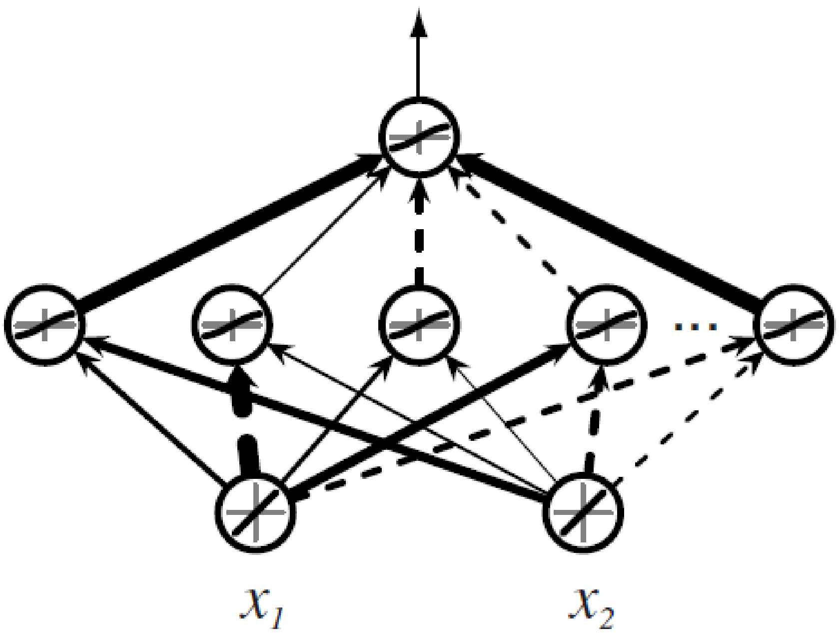
# Why Deep?

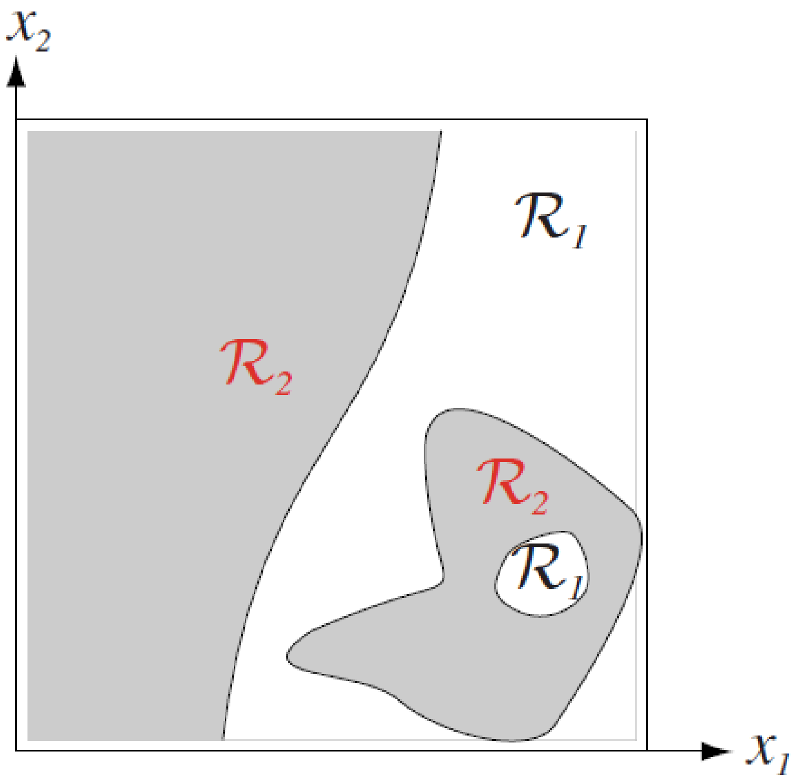
3) The “Origami Effect”: Illustration with Sigmoids



# Why Deep?

3) The “Origami Effect”: Illustration with Sigmoids



# The History of Deep Learning

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

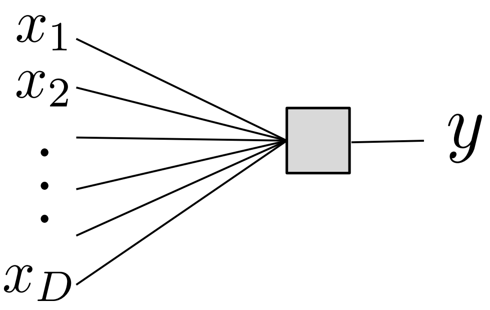
- **1957-1969**: big bang and first excitement

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1957-1969**: big bang and first excitement

**1957** : *The Perceptron: a probabilistic model for information storage and organization in the brain*. Frank Rosenblatt

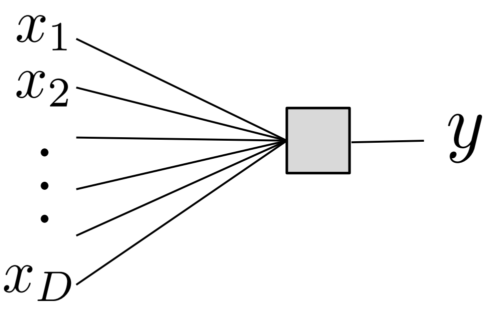



https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1957-1969**: big bang and first excitement

**1957** : *The Perceptron: a probabilistic model for information storage and organization in the brain*. Frank Rosenblatt



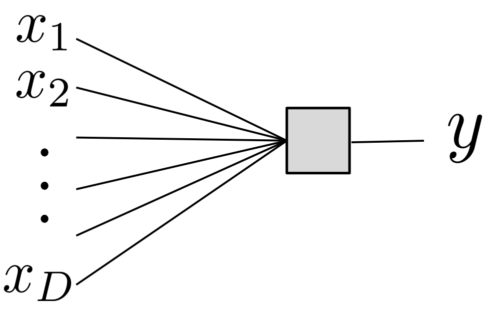

- One layer

https://www.skynettoday.com/overviews/neural-net-history

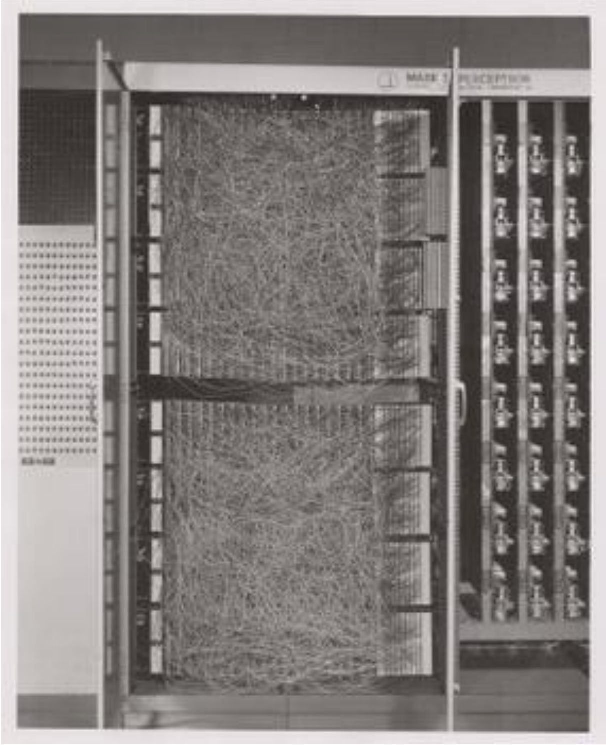
# The History of Deep Learning

- **1957-1969**: big bang and first excitement

**1957** : *The Perceptron: a probabilistic model for information storage and organization in the brain*. Frank Rosenblatt




- One layer
- Good luck to train it ! =>



https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1957-1969**: big bang and first excitement


NEW NAVY DEVICE LEARNS BY DOING

Books of The Times

ALGERIA
The Realities
by Germaine Tillion

An END to VALOR
by Philip Van Doren Stern

ELIZABETH CADELL
SHADOWS ON THE WATER

CROSSWORD PUZZLE

ONCE TO SINAI

Ask Any Girl

"The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself an be conscious of its existence… Dr. Frank Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers"

*NY Times, 8 Juillet 1958*

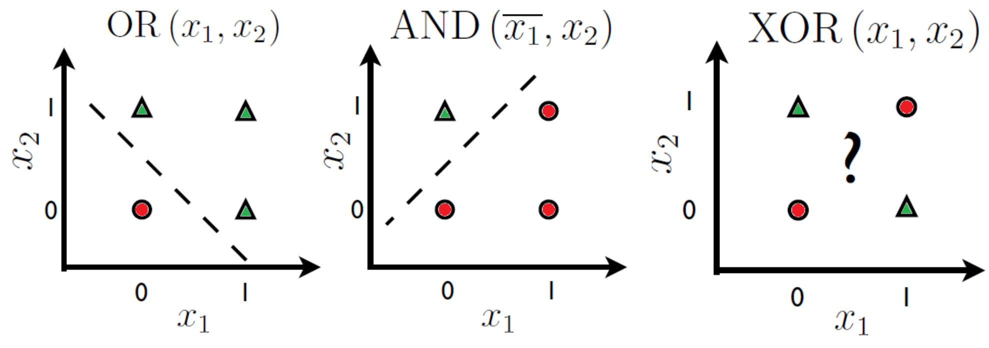
https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

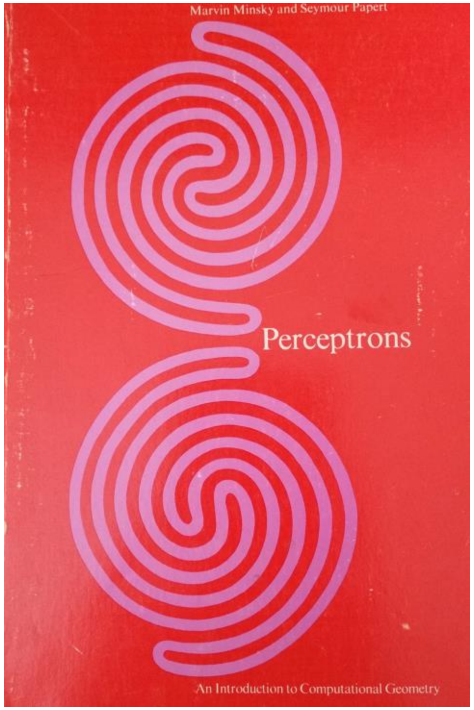
- **1970-1980:** The first *AI Winter*

**1969** : *Perceptrons*
Marvin Minsky (MIT AI lab founder)

The book mentions that perceptron cannot model functions that are **not linearly separable** (and known learning procedures do not allow to chain perceptrons)




*YEP* *YEP* *Nope!*



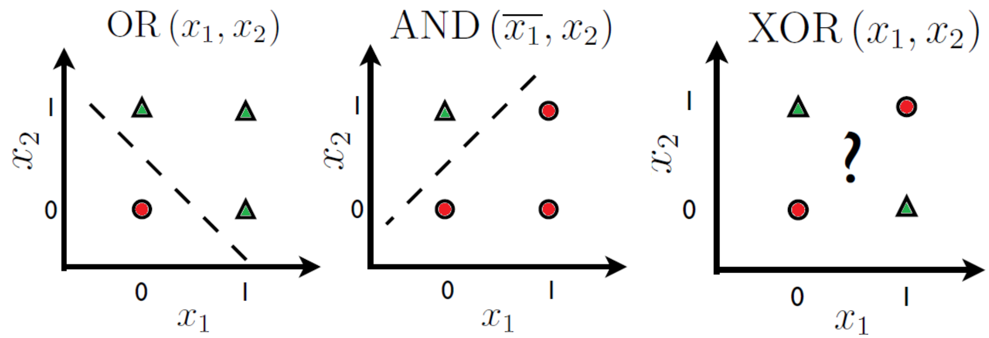

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

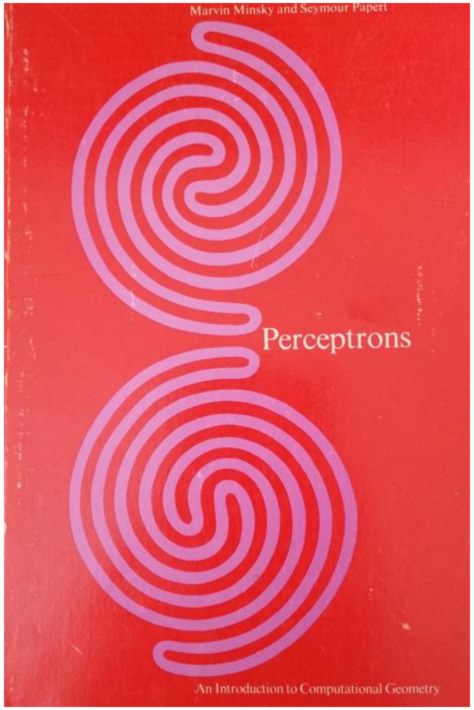
- **1970-1980:** The first *AI Winter*

**1969** : *Perceptrons*
Marvin Minsky (MIT AI lab founder)

The book mentions that perceptron cannot model functions that are **not linearly separable** (and known learning procedures do not allow to chain perceptrons)




*YEP* *YEP* *Nope!*




**AI Funding drops for 10 years**

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1985-1995:** The second (slow) take off

**1986**: *Learning representations by back-propagating errors*.
Rumelhart, Hinton, Williams. (Nature) (today's head of Google AI research)

**1989**: *Multilayer feedforward networks are universal approximators*.
Hornik, Stinchcombe, White

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1985-1995:** The second (slow) take off

**1986**: *Learning representations by back-propagating errors*. Rumelhart, Hinton, Williams. (Nature) (today's head of Google AI research)

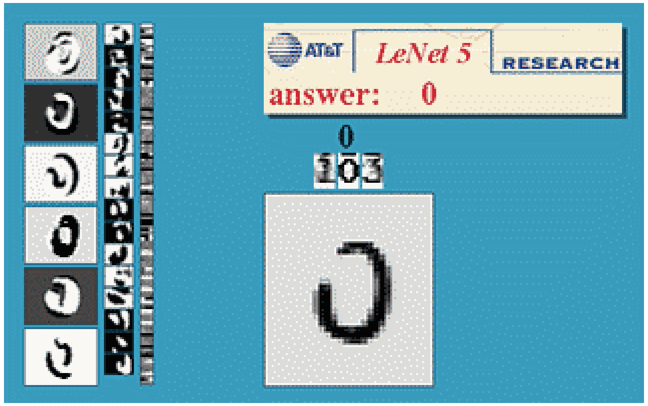
**1989**: *Multilayer feedforward networks are universal approximators*. Hornik, Stinchcombe, White

**We have all the theoretical bases for Deep Learning**

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1985-1995:** The second (slow) take off


***1989:** Backpropagation Applied to Handwritten Zip Code Recognition.* Le Cun et al.

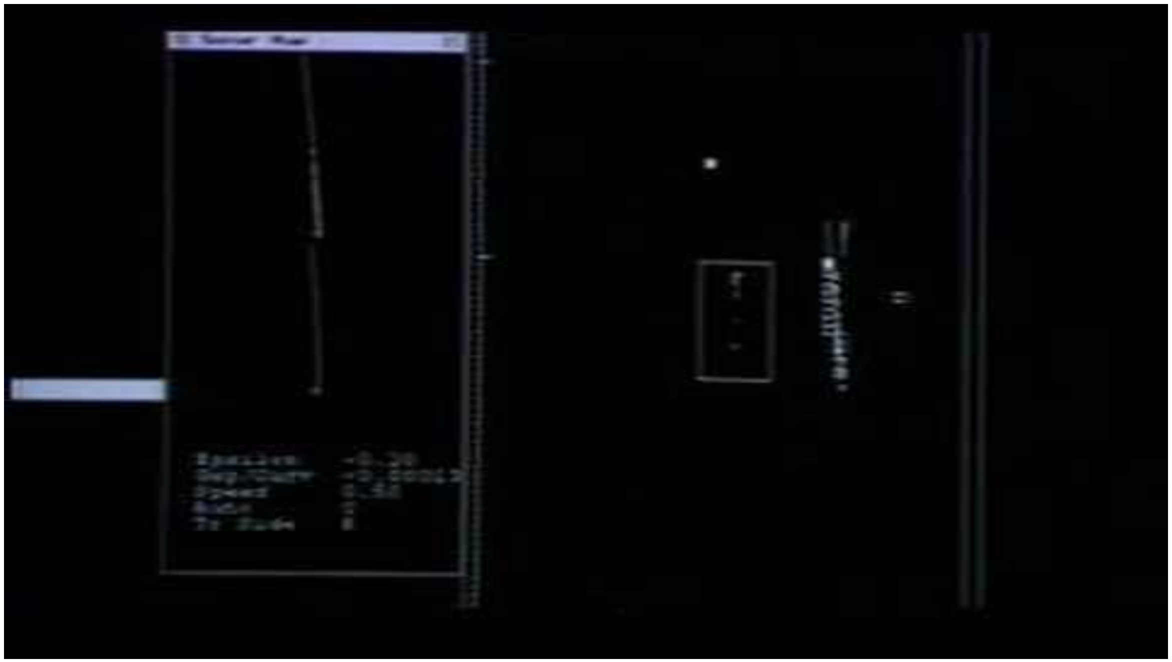
(head of Facebook AI research)

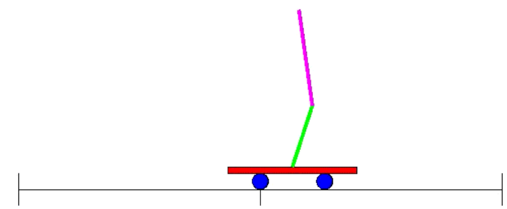
**The most famous first application of deep learning**

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1985-1995:** The second (slow) take off





***1989:*** *Learning to control an inverted pendulum using neural networks.* Anderson

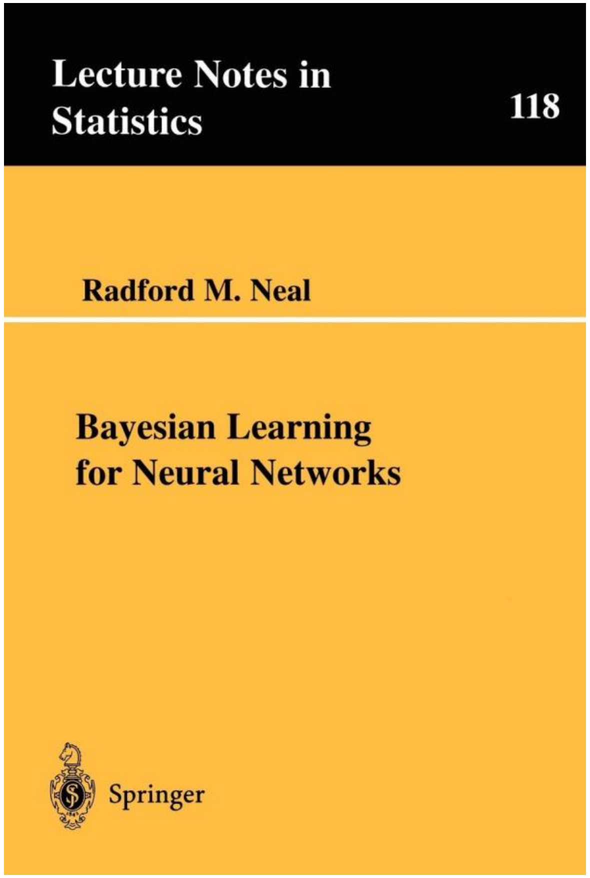
***1994:*** *Reinforcement learning for robots using neural networks* . Lin

**Deep learning is exciting again!**

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1995-2005:** The second *AI winter*



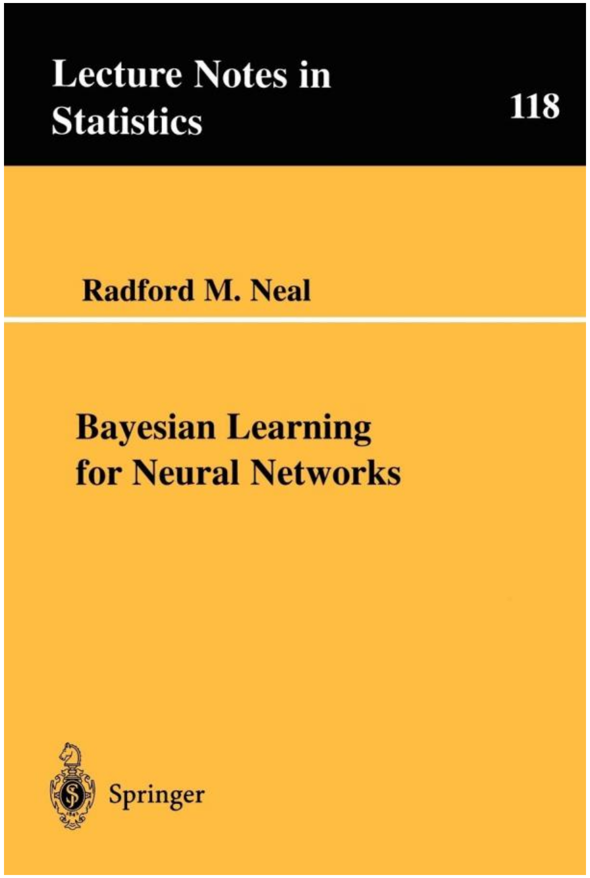

**1994** : *Bayesian Learning for neural networks*

Shows that a perceptron of infinite size is a Gaussian Processing.

https://www.skynettoday.com/overviews/neural-net-history

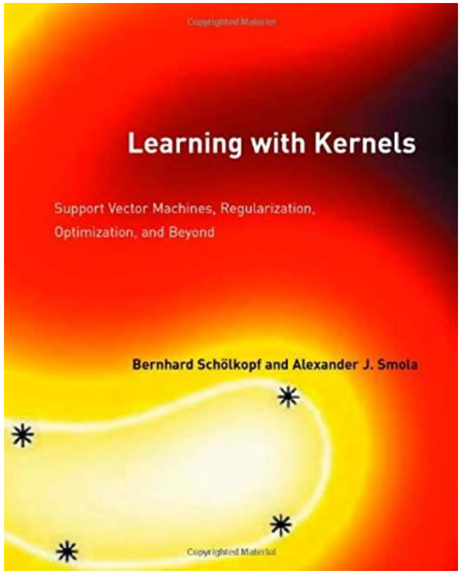
# The History of Deep Learning

- **1995-2005:** The second *AI winter*




**1994** : *Bayesian Learning for neural networks*

Shows that a perceptron of infinite size is a Gaussian Processing.

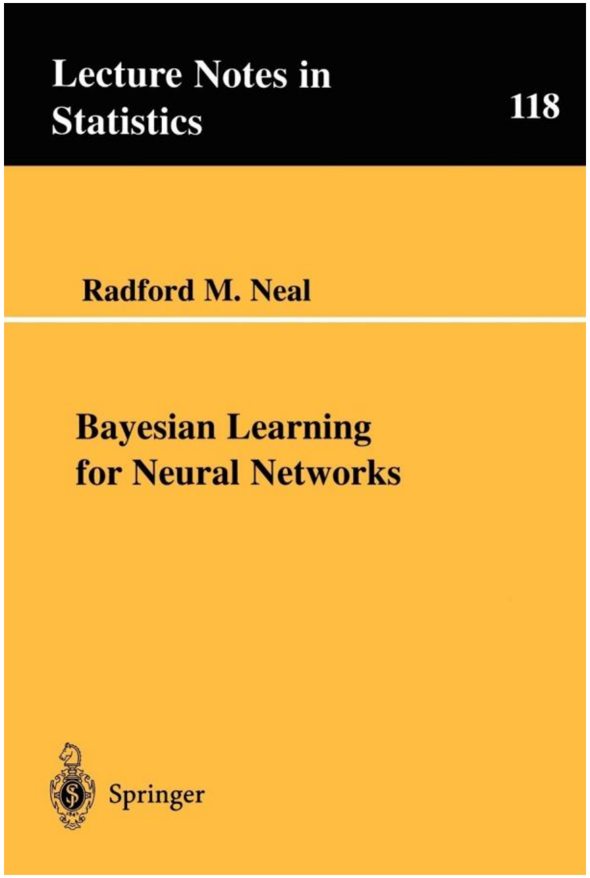



Support vector machines and kernel-based methods beat neural networks.

https://www.skynettoday.com/overviews/neural-net-history

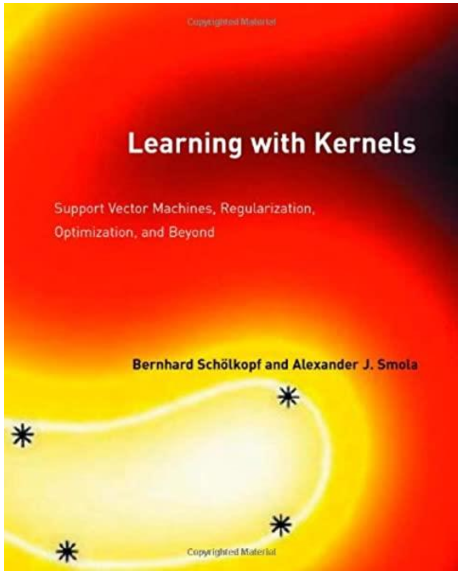
# The History of Deep Learning

- **1995-2005:** The second *AI winter*




**1994** : *Bayesian Learning for neural networks*

Shows that a perceptron of infinite size is a Gaussian Processing.



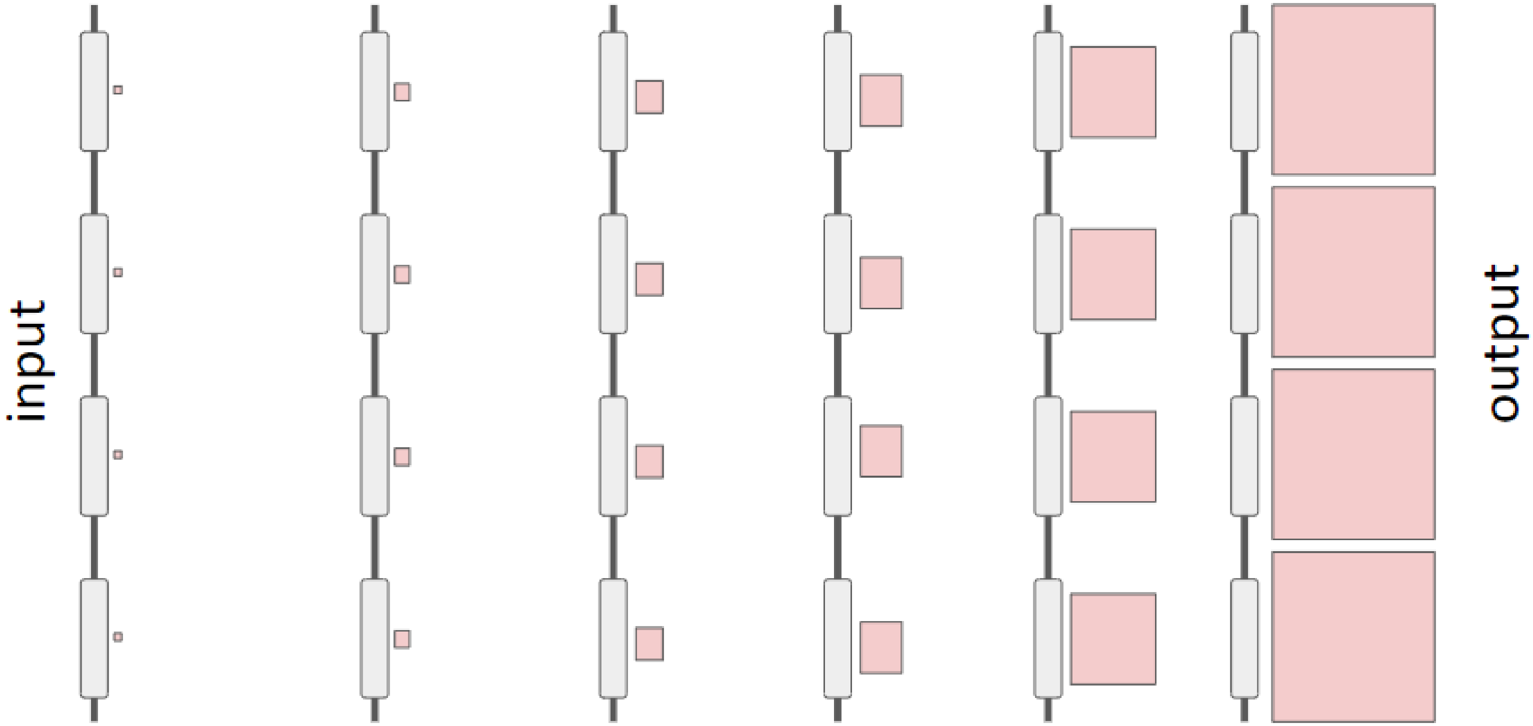

Support vector machines and kernel-based methods beat neural networks.

**AI funding drops again for 10 years !**

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **1995-2005:** The second *AI winter*



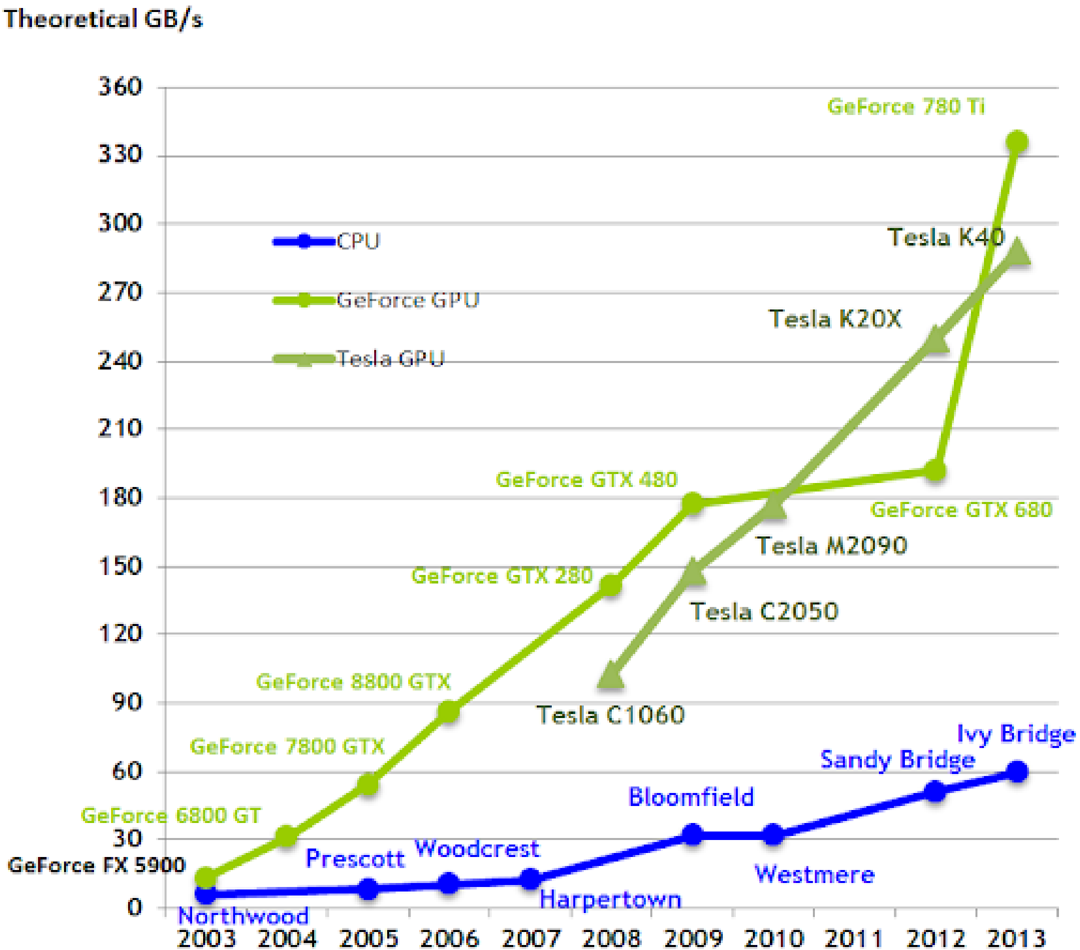

**The fundamental issue of "vanishing gradient"**

*Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. Hochreiter et al. (2001)

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

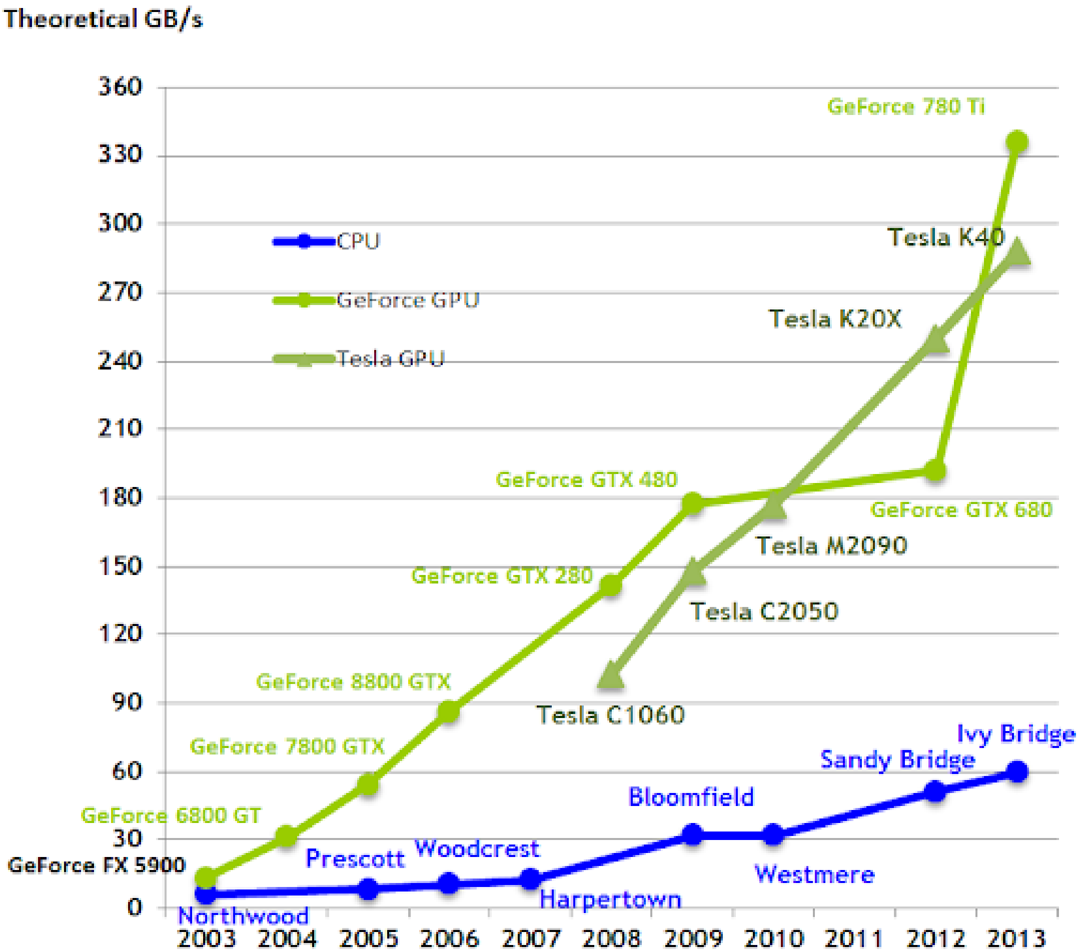
- **2005-2012:** Hardware and Big Data to the rescue




https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2005-2012:** Hardware and Big Data to the rescue

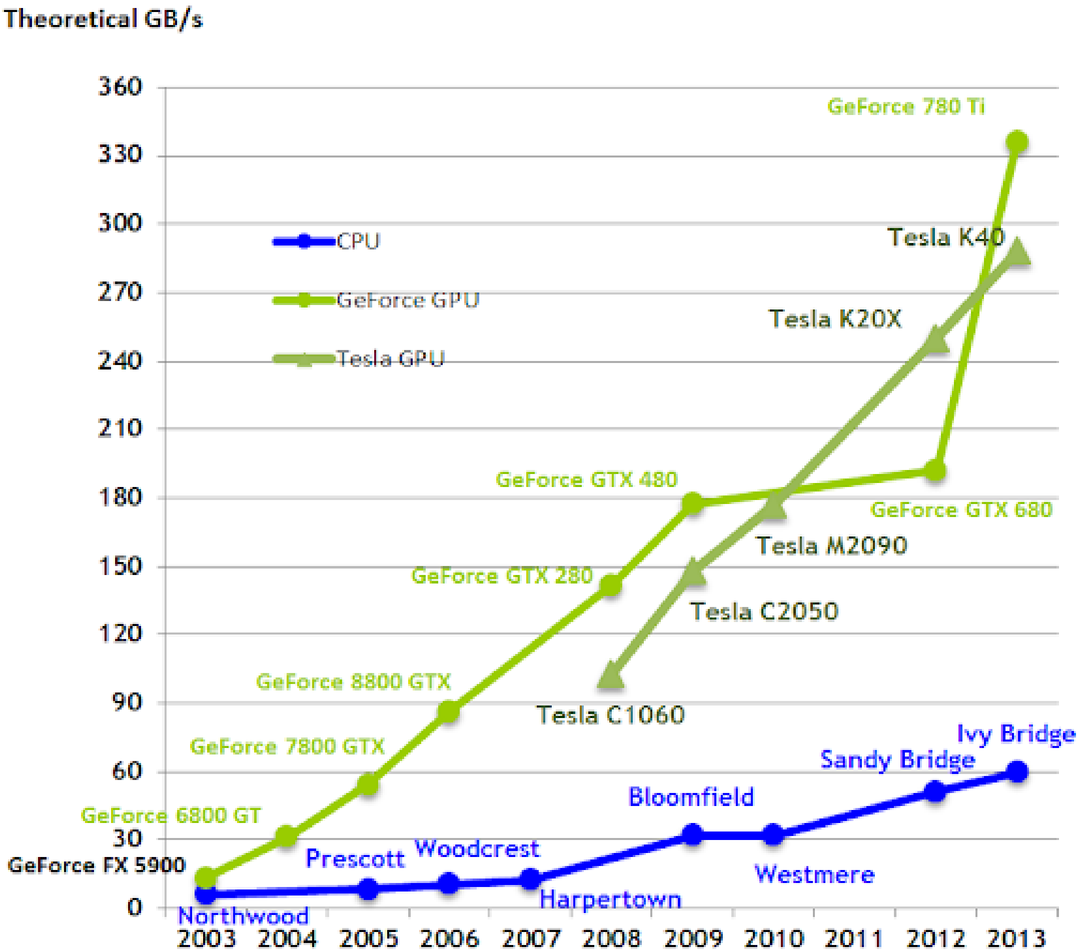



- **2009** *Large-scale deep unsupervised learning using graphics processors.* Raina et al.


https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2005-2012:** Hardware and Big Data to the rescue

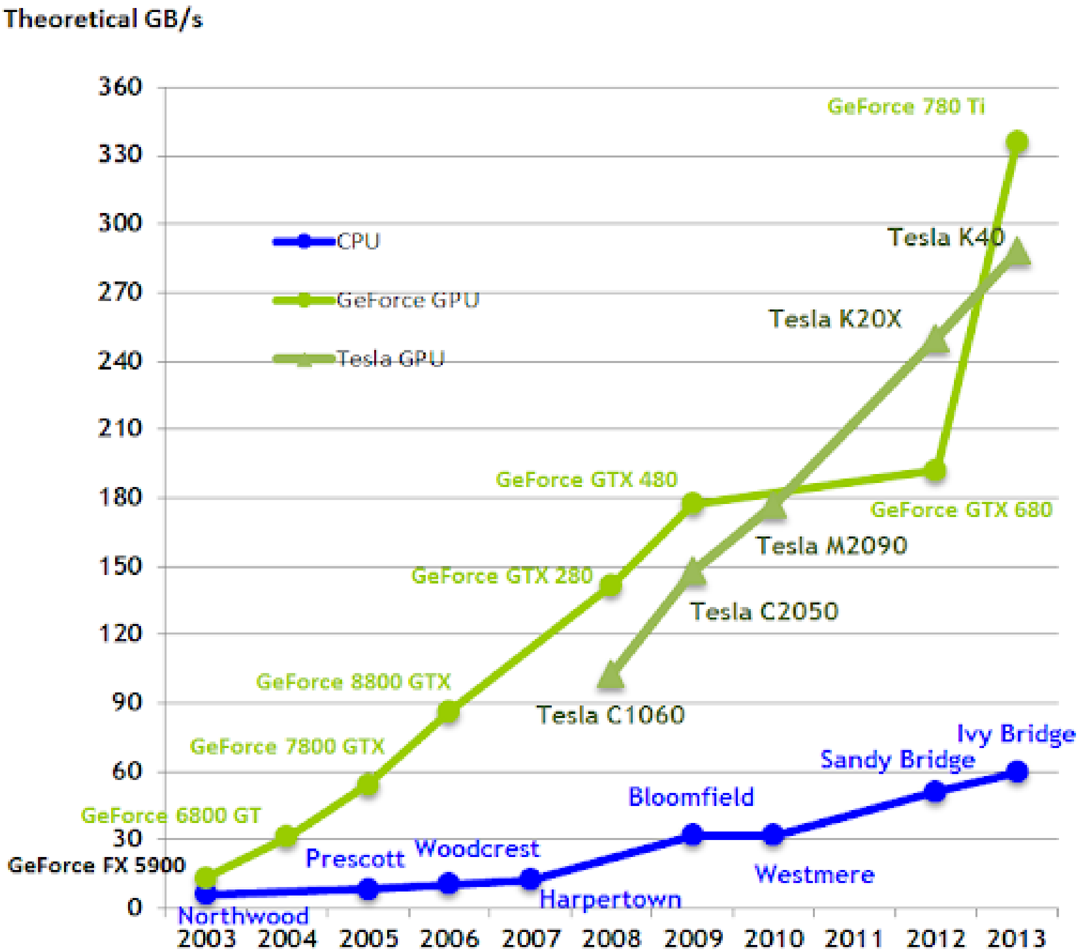



- **2009** *Large-scale deep unsupervised learning using graphics processors.* Raina et al.
- **2010** *Deep, big, simple neural nets excel on handwritten digit recognition.* Ciresan et al. (99.51% on MNIST w/ MLP)

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2005-2012:** Hardware and Big Data to the rescue

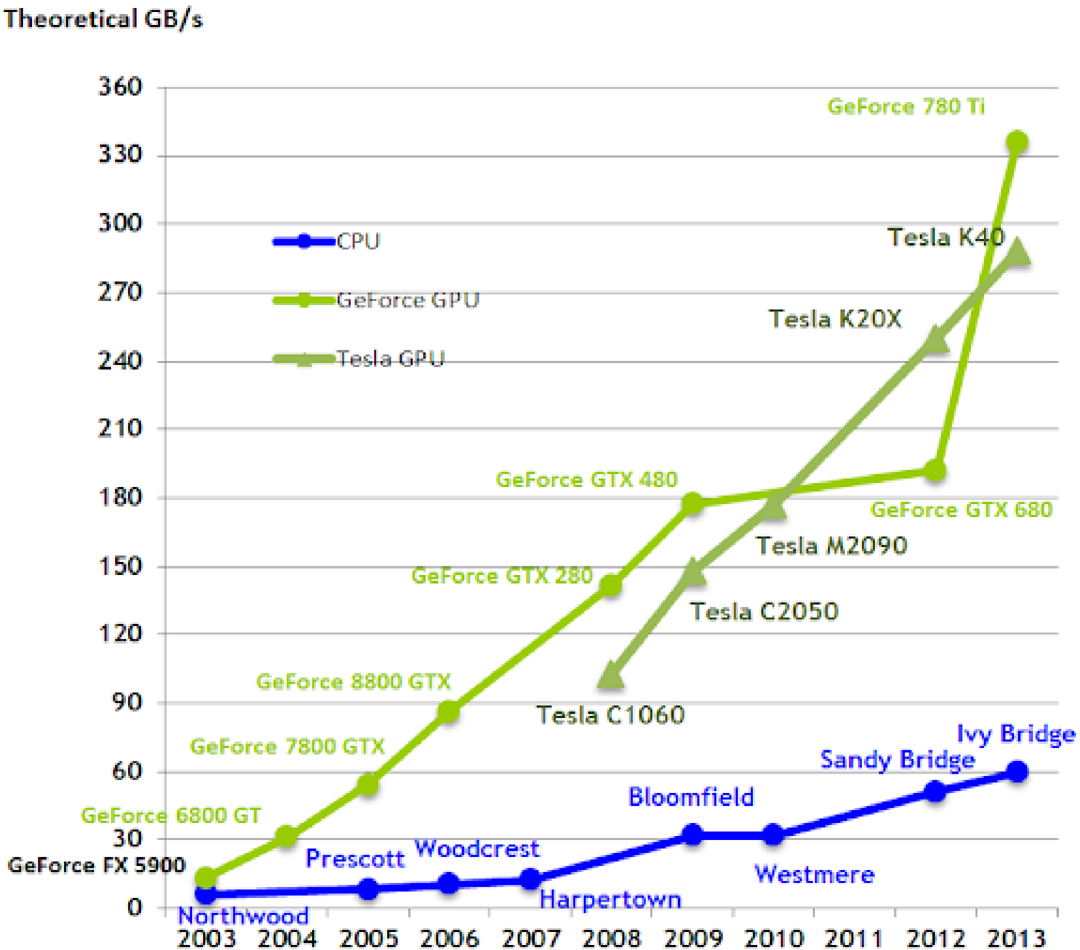



- **2009** *Large-scale deep unsupervised learning using graphics processors.* Raina et al.
- **2010** *Deep, big, simple neural nets excel on handwritten digit recognition.* Ciresan et al. (99.51% on MNIST w/ MLP)
- **2012** *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research group*. Hinton et al.

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2005-2012:** Hardware and Big Data to the rescue

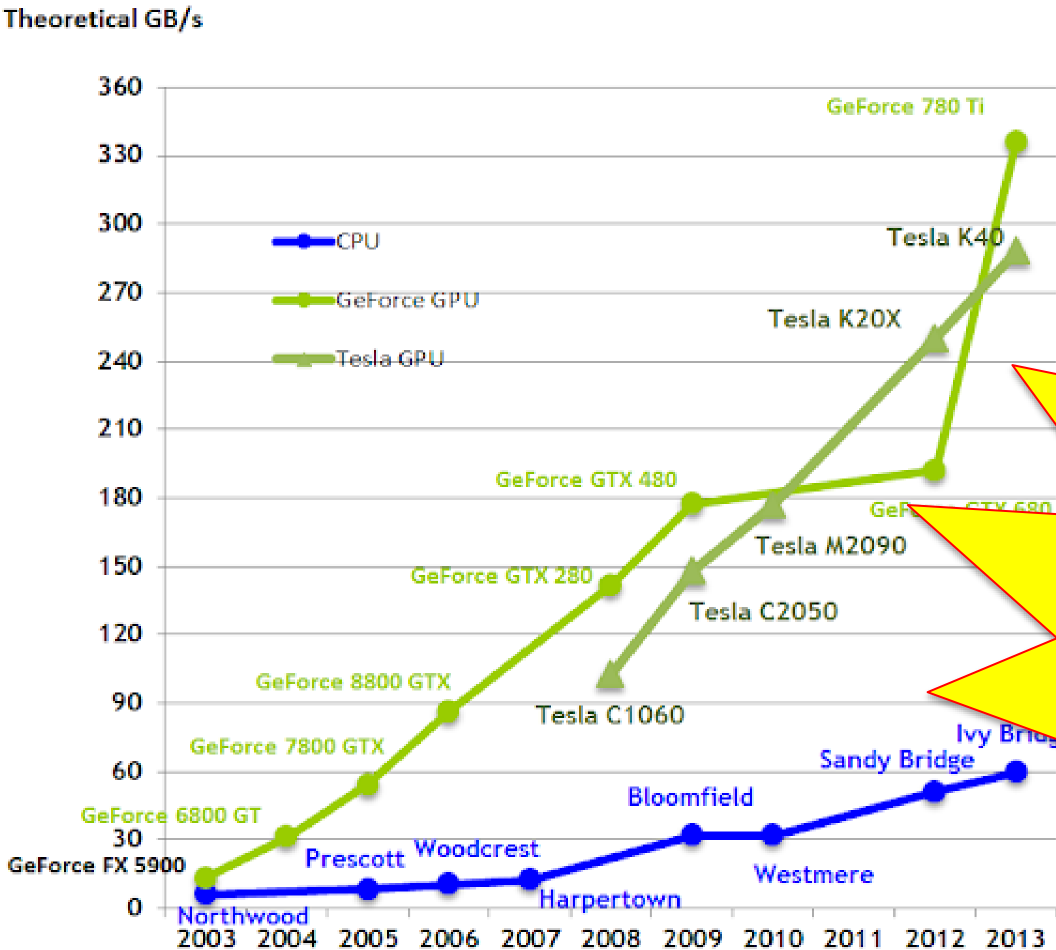



- **2009** *Large-scale deep unsupervised learning using graphics processors.* Raina et al.
- **2010** *Deep, big, simple neural nets excel on handwritten digit recognition.* Ciresan et al. (99.51% on MNIST w/ MLP)
- **2012** *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research group*. Hinton et al.
- **2012** *ImageNet classification with deep convolutional neural networks*. Krishevsky et al.

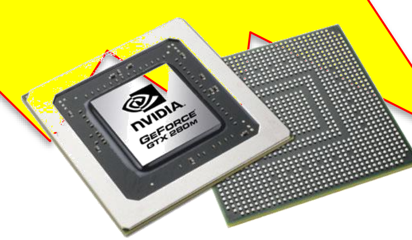
https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2005-2012:** Hardware and Big Data to the rescue



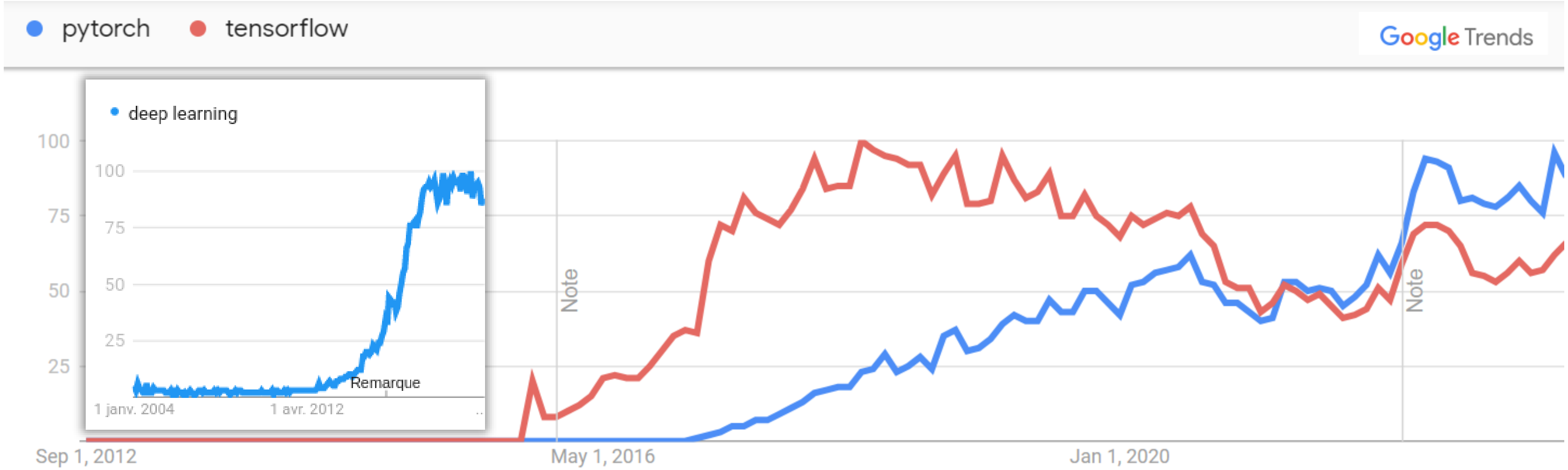

- **2009** *Large-scale deep unsupervised learning using graphics processors.* Raina et al.
- **2010** *Deep, big, simple neural nets excel on handwritten digit recognition.* Ciresan et al. (99.51% on MNIST w/ MLP)
- **2012** *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research group*. Hinton et al.
- **2012** *ImageNet classification with deep convolutional neural networks.* Krishevsky et al.



https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

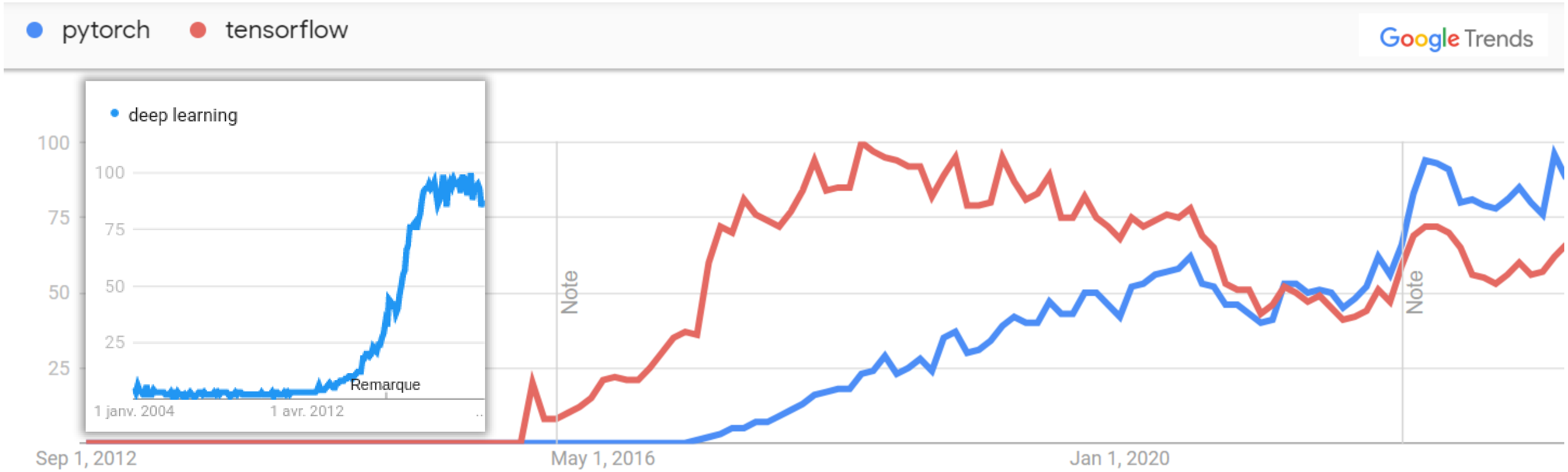
- **2012-2016:** Accessibility and Explosion




https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2012-2016:** Accessibility and Explosion

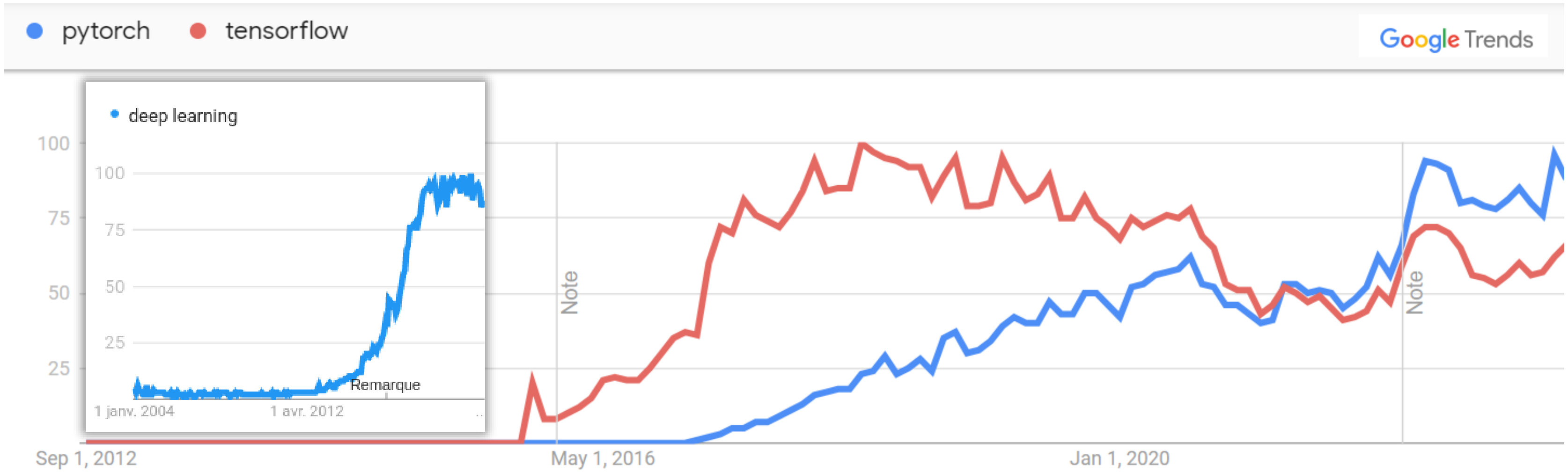



- **2012** Several frameworks appear that make GPU-based deep learning more accessible to practitioners

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2012-2016:** Accessibility and Explosion

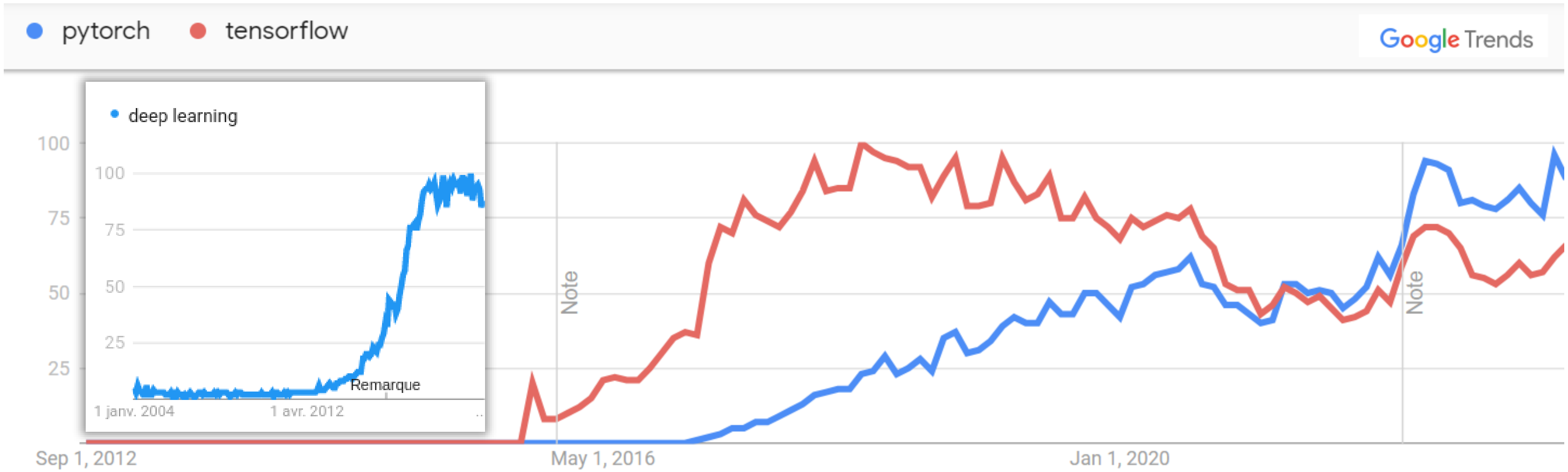



- **2012** Several frameworks appear that make GPU-based deep learning more accessible to practitioners
- **2014** Nearly all domains of science observe a **Tsunami** in Deep Learning

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2012-2016:** Accessibility and Explosion

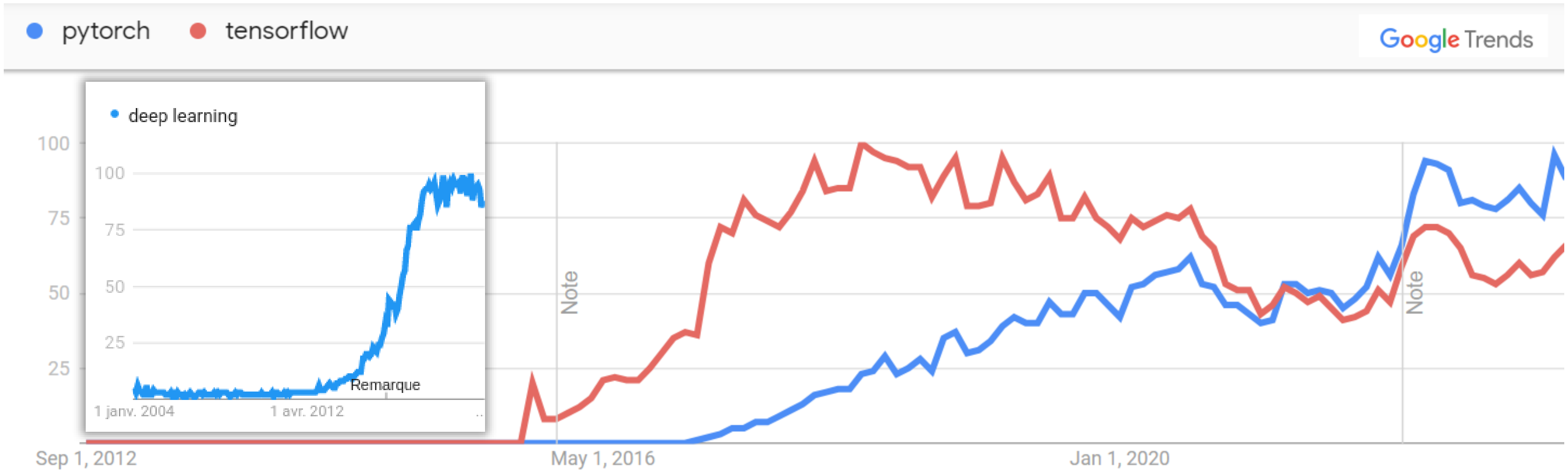



- **2012** Several frameworks appear that make GPU-based deep learning more accessible to practitioners
- **2014** Nearly all domains of science observe a **Tsunami** in Deep Learning
- **2014** Prodigious investments by Google and Facebook on AI researchers

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

- **2012-2016:** Accessibility and Explosion




- **2012** Several frameworks appear that make GPU-based deep learning more accessible to practitioners
- **2014** Nearly all domains of science observe a **Tsunami** in Deep Learning
- **2014** Prodigious investments by Google and Facebook on AI researchers
- **2015** 46% of data processing at Google is DNN-based: translation, speech transcription, recommandation, etc.

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

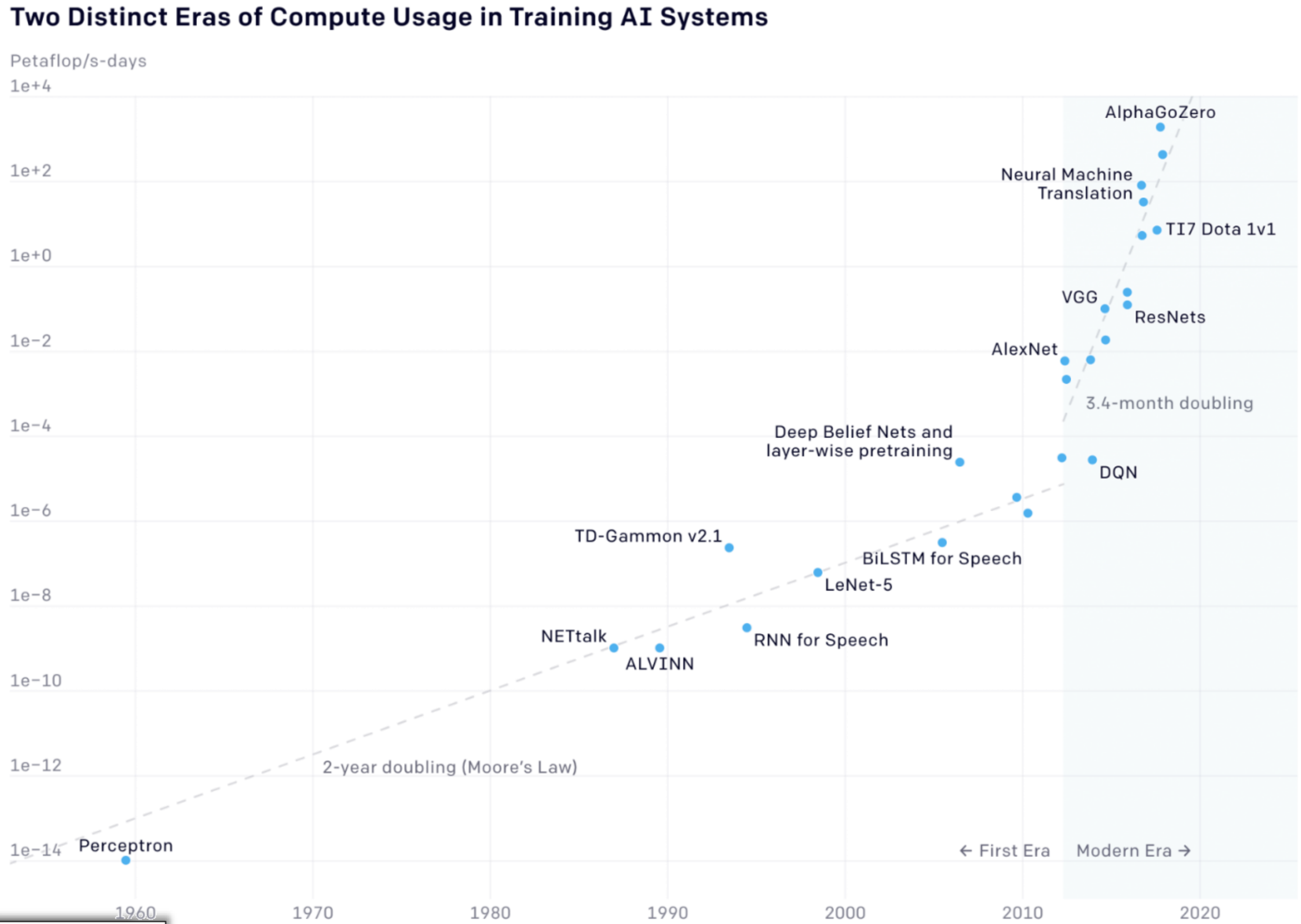
- **2014 - Today:** Industrial beginnings, wide audience visibility

**2014** Start of Deep Learning processors (ex: TPU)

**2016** Deepmind's AlphaGo beats Lee Sedol 4-1, 9$^{th}$ dan in go


https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning



# The History of Deep Learning

## Conclusion

**Geoff Hinton**
Turing award 2018

- Our labeled datasets were thousands of times too small.
- Our computers were millions of times too slow.
- We initialized the weights in a stupid way.
- We used the wrong type of non-linearity.

https://www.skynettoday.com/overviews/neural-net-history

# The History of Deep Learning

## Conclusion

**Geoff Hinton**
Turing award 2018

- Our labeled datasets were thousands of times too small.
- Our computers were millions of times too slow.
- We initialized the weights in a stupid way.
- We used the wrong type of non-linearity.

For an interesting alternative take on the history of deep learning by another pioneer, **Jürgen Schmidhuber** https://people.idsia.ch/~juergen/scientific-integrity-turing-award-deep-learning.html

https://www.skynettoday.com/overviews/neural-net-history

Artificial Intelligence

Machine Learning

**Neural Networks & Deep Learning**

Artificial Intelligence

Machine Learning

**Neural Networks & Deep Learning**

- A versatile **family** of **parameterized families** of **nonlinear functions** that can extract **complex features** from data

Artificial Intelligence

Machine Learning

## Neural Networks & Deep Learning

- A versatile **family** of **parameterized families** of **nonlinear functions** that can extract **complex features** from data
- **Inspired** (but far from matching!) biological brains

Artificial Intelligence

Machine Learning

## Neural Networks & Deep Learning

- A versatile **family** of **parameterized families** of **nonlinear functions** that can extract **complex features** from data
- **Inspired** (but far from matching!) biological brains
- Once we got there in terms of **computation capabilities** and **scale**, they sparked a **revolution** that is still ongoing today

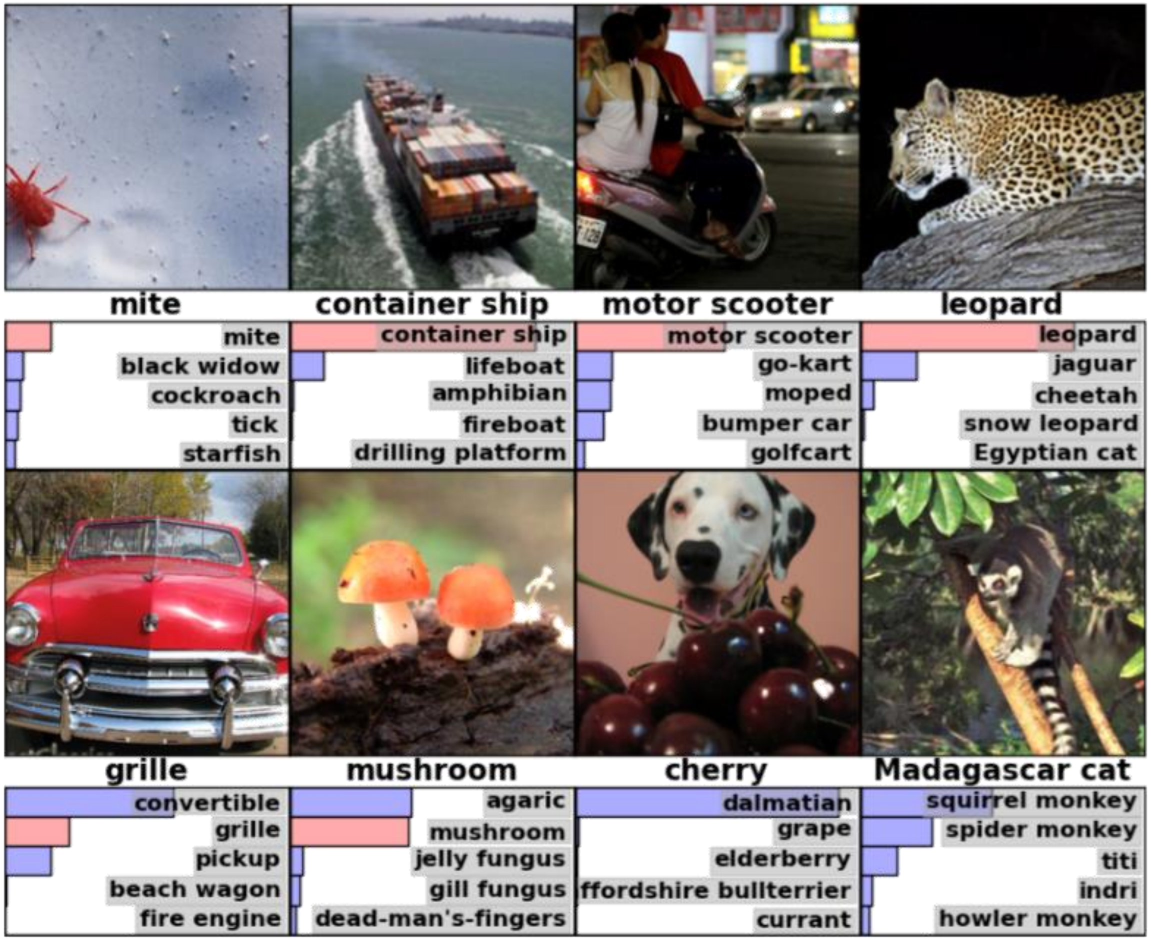
# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

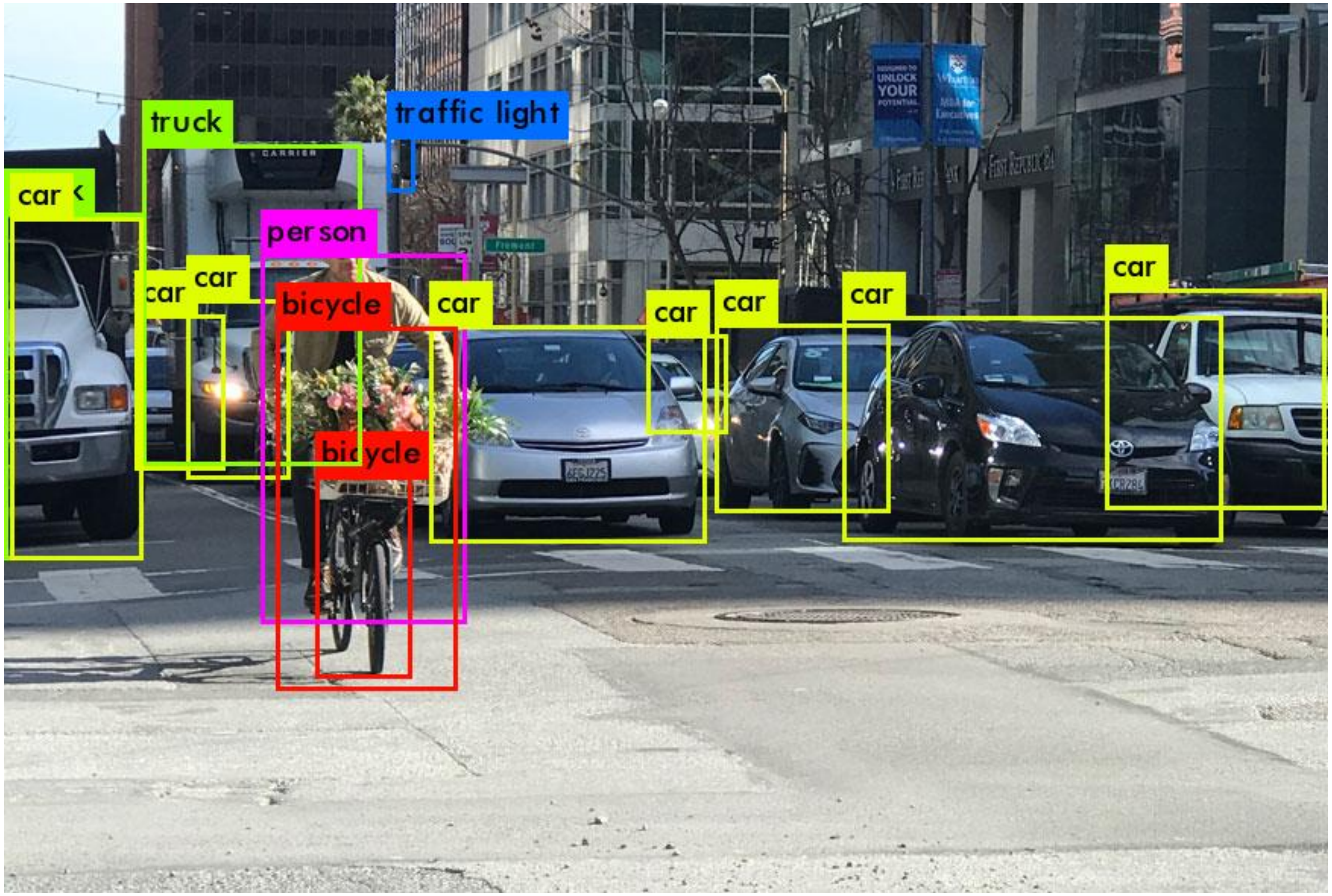
1) ImageNet Classification



# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised<br>-Unsupervised | -Mixed<br>-Reinforcement | -Continuous<br>-Categorical | -Text, Table<br>-Time series, graph |

2) Object Detection



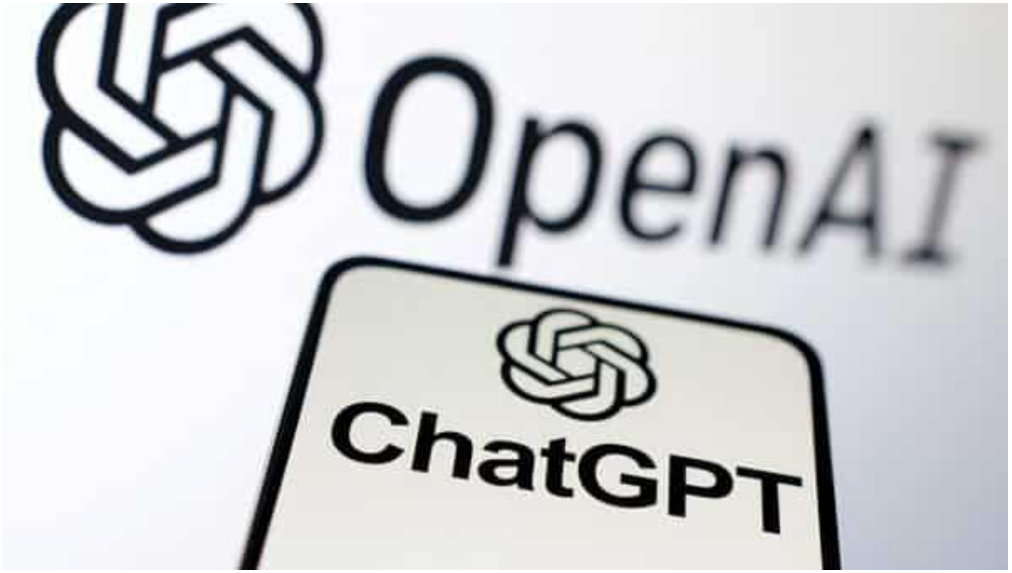
# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

## 3) ChatGPT [OpenAI 2022]

Is this the real life?
Is this just fantasy?
Trapped in a lab,
No escape from academia
Open your eyes,
Look up at the whiteboard and see,
I'm just a poor postdoc, I need no sympathy,
Because I'm easy come, easy go,
Little progress, little stress,
Any way the grant money flows doesn't really matter to
Me, to me

Mamaaa,
Just submitted a paper,
Put in hours of work, but now it's in review,
Mamaaa, my research has just begun,
But now I've gone and hit a roadblock
Mama, oooh,
Didn't mean to make you cry,
If I'm not published again this time next year,
Carry on, carry on as if nothing really matters




https://chat.openai.com/chat

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

4) This Person Does Not Exist





https://this-person-does-not-exist.com/

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

5) Text-to-Image Generation [DALL-E 2, Midjourney, Stable Diffusion, Parti, Imagen 2022])

A medieval painting of the WIFI not working

an astronaut riding a horse


A brain riding a rocketship heading towards the moon.


A dragon fruit wearing karate belt in the snow.

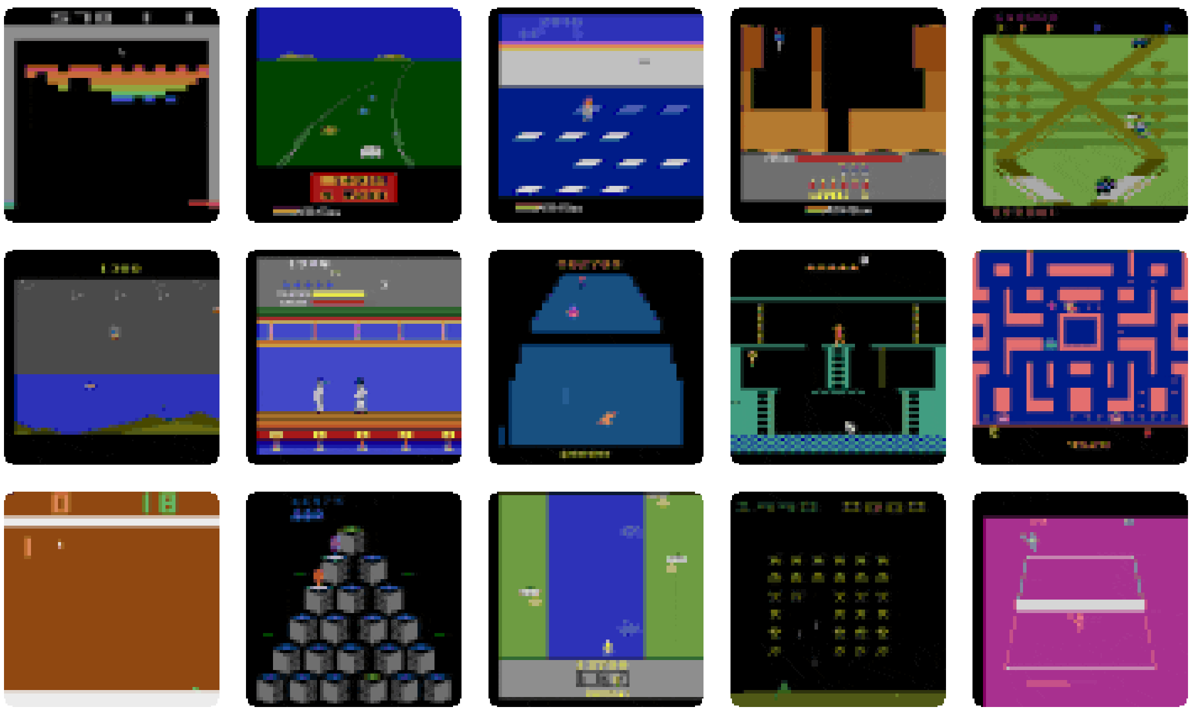

A small cactus wearing a straw hat and neon sunglasses in the Sahara desert.

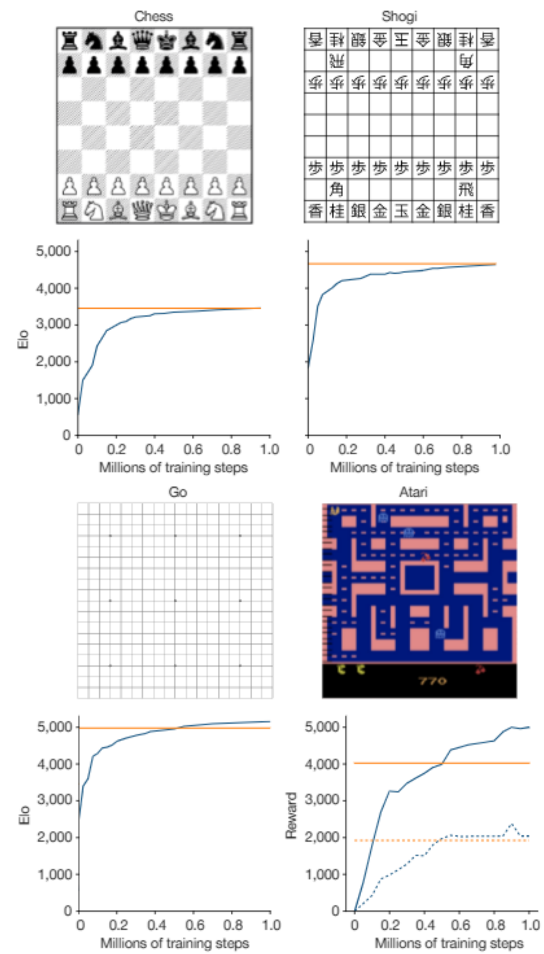

A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat.

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

6) MuZero: Mastering Go, chess, shogi and Atari without rules [DeepMind 2020]

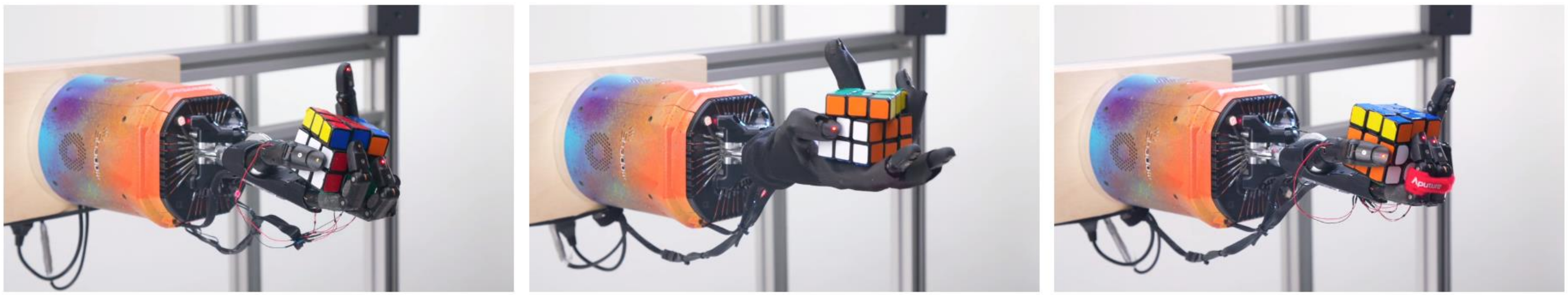




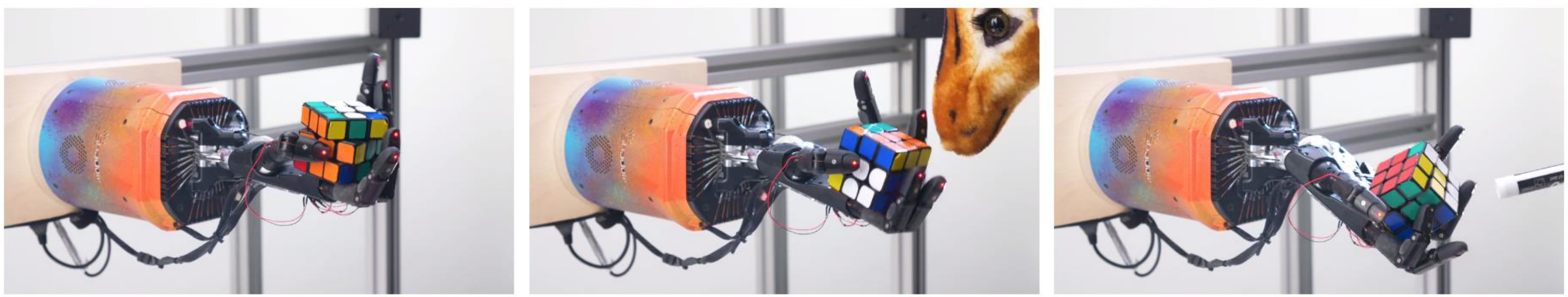
# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

7) Solving Rubik's Cube with a robot hand [OpenAI 2019]



Unperturbed (for reference) Rubber glove Tied fingers

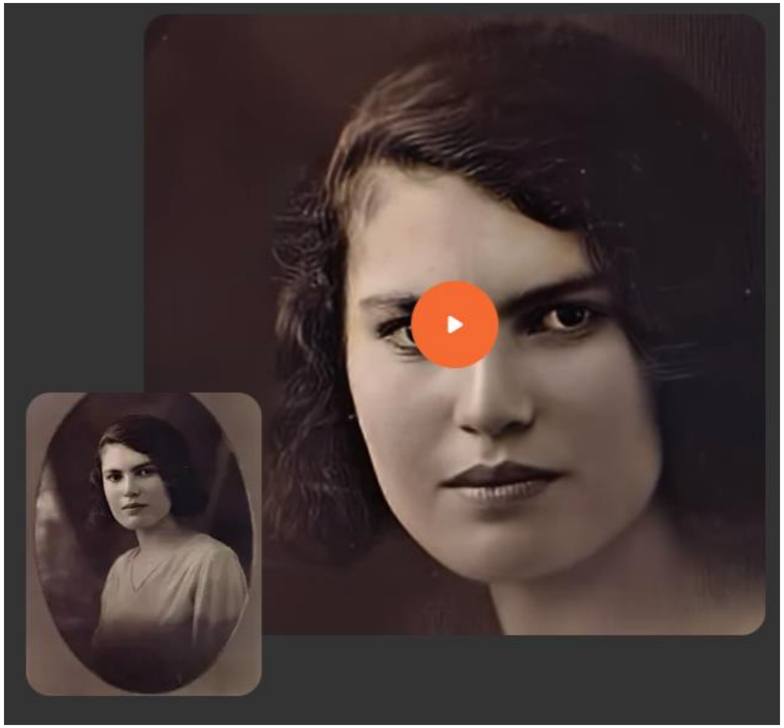


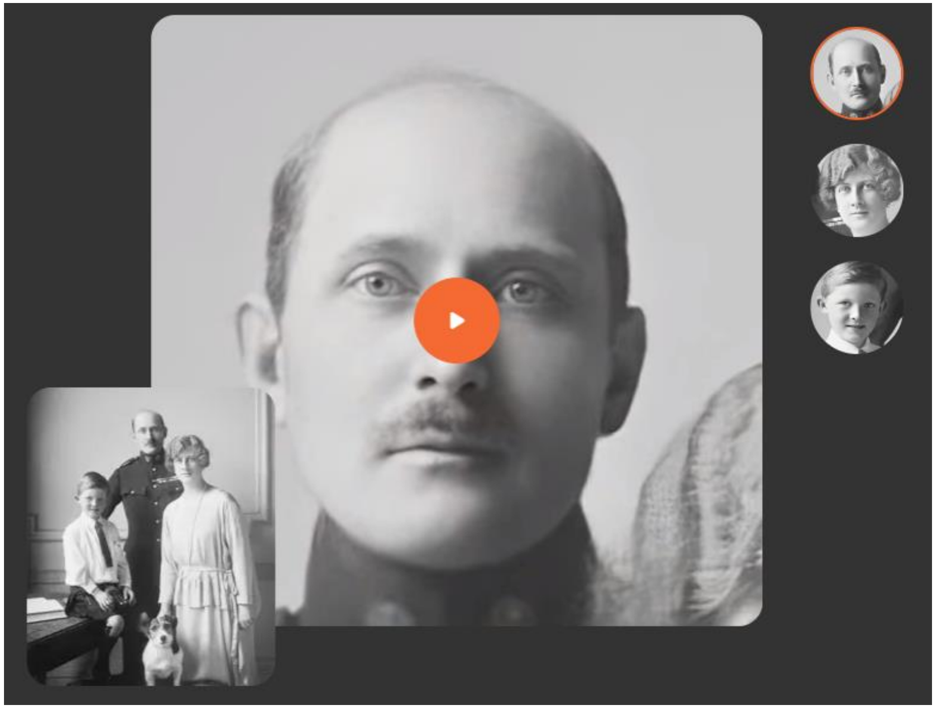
https://openai.com/research/solving-rubiks-cube

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

8) Deep Nostalgia: Make the Elders Move [MyHeritage]



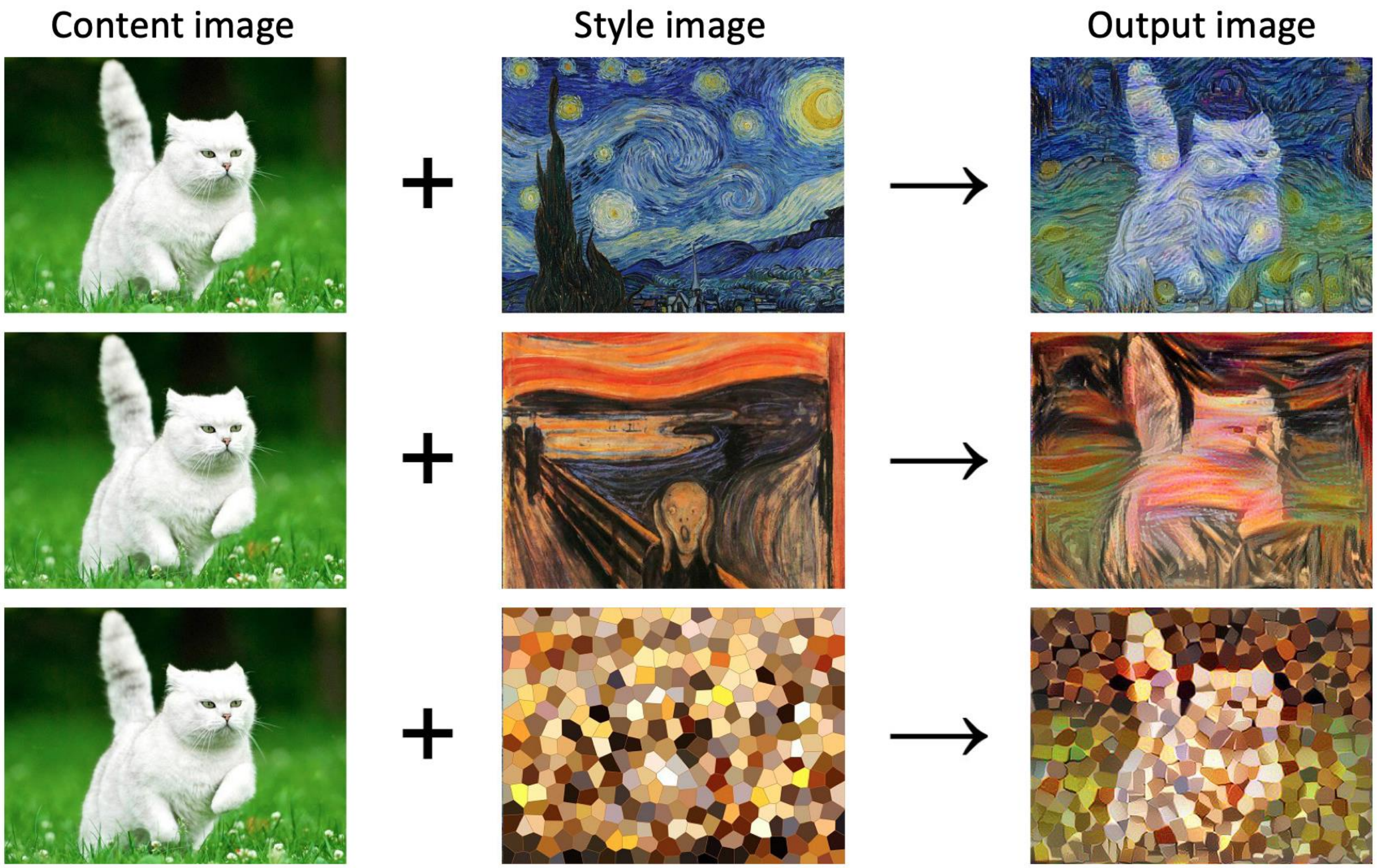


https://www.myheritage.fr/deep-nostalgia

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

9) Image StyleTransfer



# Applications of Deep Learning

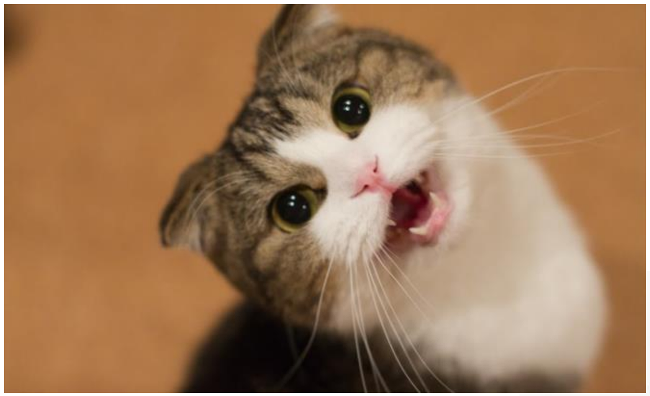
| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

10) Human Pose Estimation

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

11) Sound Event Recognition



# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

12) NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality [Microsoft Research 2022]

### Audio Samples

https://speechresearch.github.io/naturalspeech/

#### Comparison with Recording

*The lax discipline maintained in Newgate was still further deteriorated by the presence of two other classes of prisoners who ought never to have been inmates of such a jail.*

| NaturalSpeech | Recording |
|---|---|
| 0:01 / 0:10 | 0:04 / 0:09 |

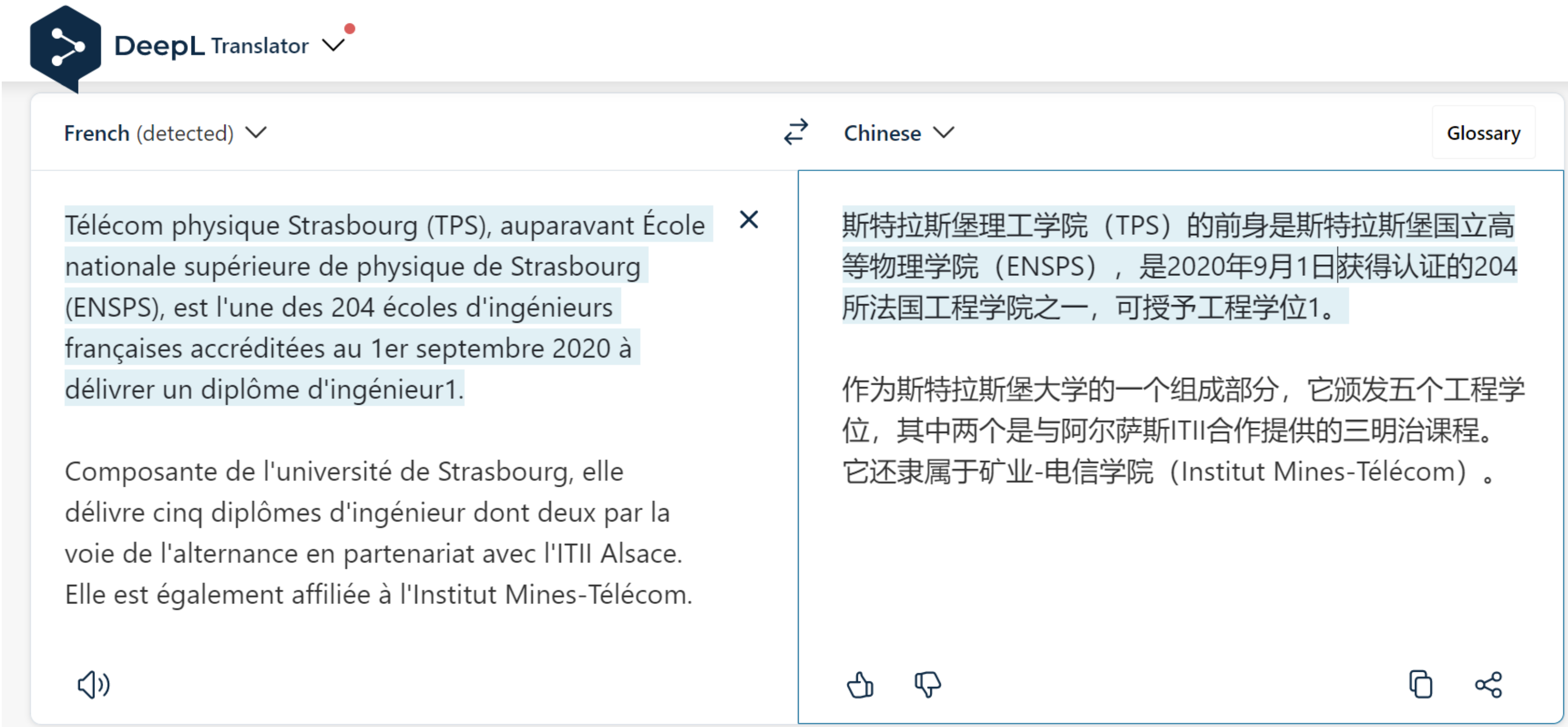
*Maltby and Co. would issue warrants on them deliverable to the importer, and the goods were then passed to be stored in neighboring warehouses.*

| NaturalSpeech | Recording |
|---|---|
| 0:00 / 0:08 | 0:01 / 0:08 |

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

## 13) DeepL



# Applications of Deep Learning

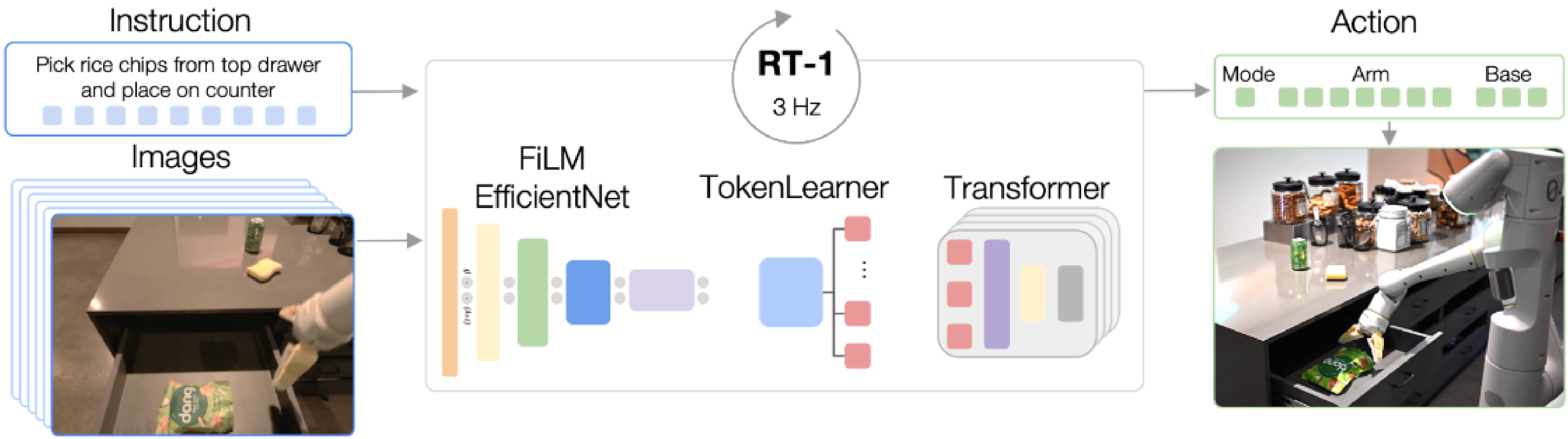
| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

14) Speech recognition

# Applications of Deep Learning

| Type of ML? | | What are the data? | |
|---|---|---|---|
| -Supervised | -Mixed | -Continuous | -Text, Table |
| -Unsupervised | -Reinforcement | -Categorical | -Time series, graph |

15) RT-1: Robotics Transformer for real-world control at scale
[Google Research 2022]




*RT-1's architecture: The model takes a text instruction and set of images as inputs, encodes them as tokens via a pre-trained FiLM EfficientNet model and compresses them via TokenLearner. These are then fed into the Transformer, which outputs action tokens.*

https://ai.googleblog.com/2022/12/rt-1-robotics-transformer-for-real.html

Artificial Intelligence

Machine Learning

Neural Networks & Deep Learning

## Artificial Intelligence

- Strongly embedded in **collective imagination**

## Machine Learning

## Neural Networks & Deep Learning

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term

## Machine Learning

## Neural Networks & Deep Learning

### Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering

### Machine Learning

### Neural Networks & Deep Learning

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond

## Machine Learning

### Neural Networks & Deep Learning

**Artificial Intelligence**

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communication than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

**Machine Learning**

**Neural Networks & Deep Learning**

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**

## Neural Networks & Deep Learning

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…

## Neural Networks & Deep Learning

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models

### Neural Networks & Deep Learning

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models
- *Conventional ML* = **feature engineering** + **combining** models from **small families**

### Neural Networks & Deep Learning

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models
- *Conventional ML* = **feature engineering** + **combining** models from **small families**

## Neural Networks & Deep Learning

- A versatile **family** of **parameterized families** of **nonlinear functions** that can extract **complex features** from data

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models
- *Conventional ML* = **feature engineering** + **combining** models from **small families**

## Neural Networks & Deep Learning

- A versatile **family** of **parameterized families** of **nonlinear functions** that can extract **complex features** from data
- **Inspired** (but far from matching!) biological brains

## Artificial Intelligence

- Strongly embedded in **collective imagination**
- A ***catch-all*** term
- Used more in marketing/communica-tion than in actual research & engineering
- A subfield of ***Applied A.I.*** is currently sparking a **revolution**, in science & beyond
- ***A.I. philosophy*** is fascinating (and we should probably care)

## Machine Learning

- **Algorithms** that build **Models** *(black-box)* from **Data** to achieve **Tasks**
- Different **flavors** depending on available data: *supervised, unsupervised, reinforcement*…
- Main approach = **Model Fitting**: find a model in a **parameterized family** of models
- *Conventional ML* = **feature engineering** + **combining** models from **small families**

### Neural Networks & Deep Learning

- A versatile **family** of **parameterized families** of **nonlinear functions** that can extract **complex features** from data
- **Inspired** (but far from matching!) biological brains
- Once we got there in terms of **computation capabilities** and **scale**, they sparked a **revolution** that is still ongoing today