Deep Learning for Data Science
DS598 B1

https://dl4ds.github.io/sp2024/

Introduction and Course Overview

Staff

Thomas Gardos
Instructor

tgardos@bu.edu
@trgardos
thomas-gardos
trgardos

Xavier Thomas
Teaching Assistant

xthomas@bu.edu
xavierohan

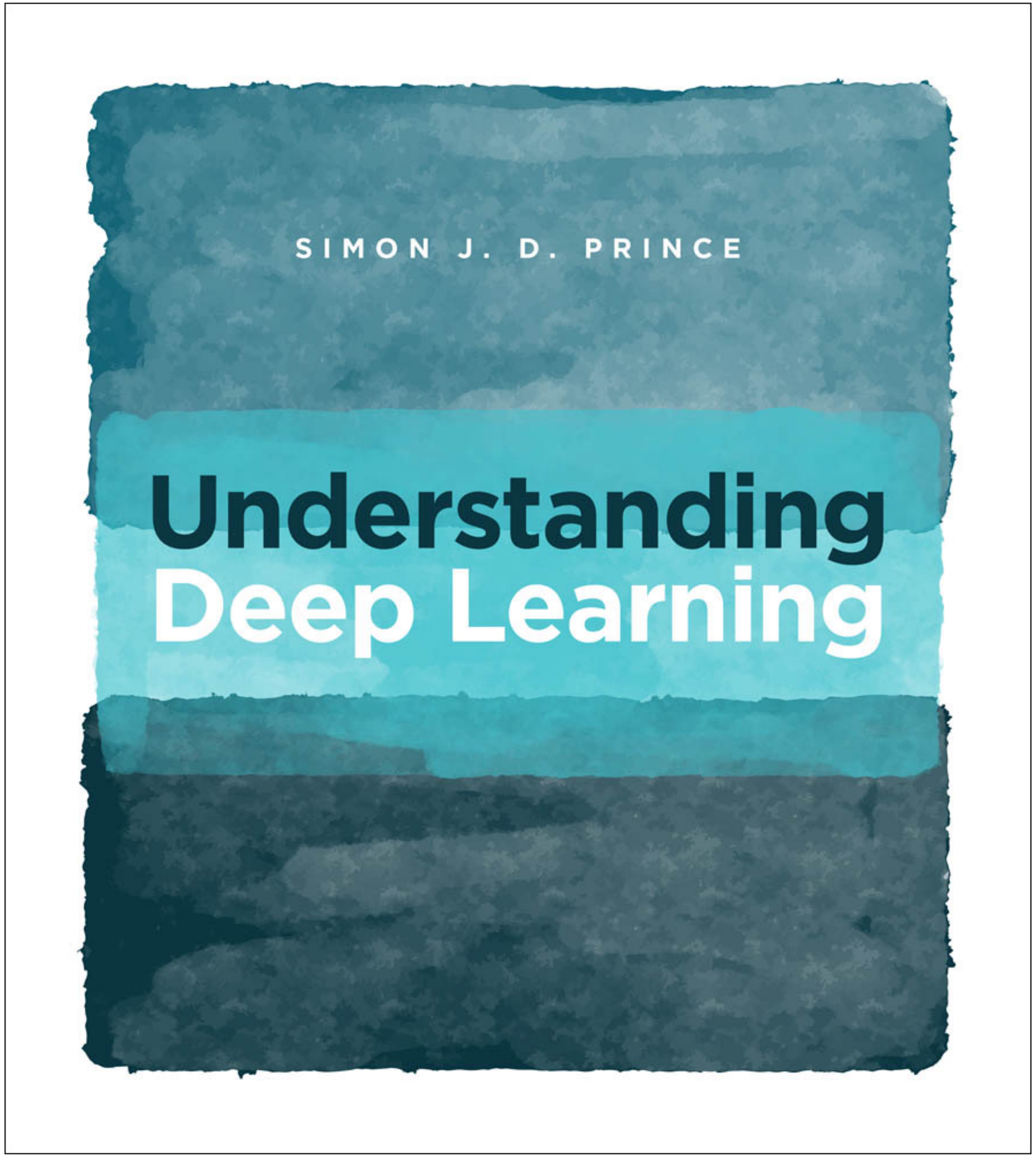
Terrier Tutor

OpenAI GPT
(Subscription Required)

GitHub Repo
(Under Development)

Book

- Published December 2023
- http://udlbook.com
  - Free PDF there or buy at BU bookstore
  - Jupyter Notebooks (we'll be revising)
  - Problem Sets
- Used heavily for 1st half of the course, and a bit at the end too



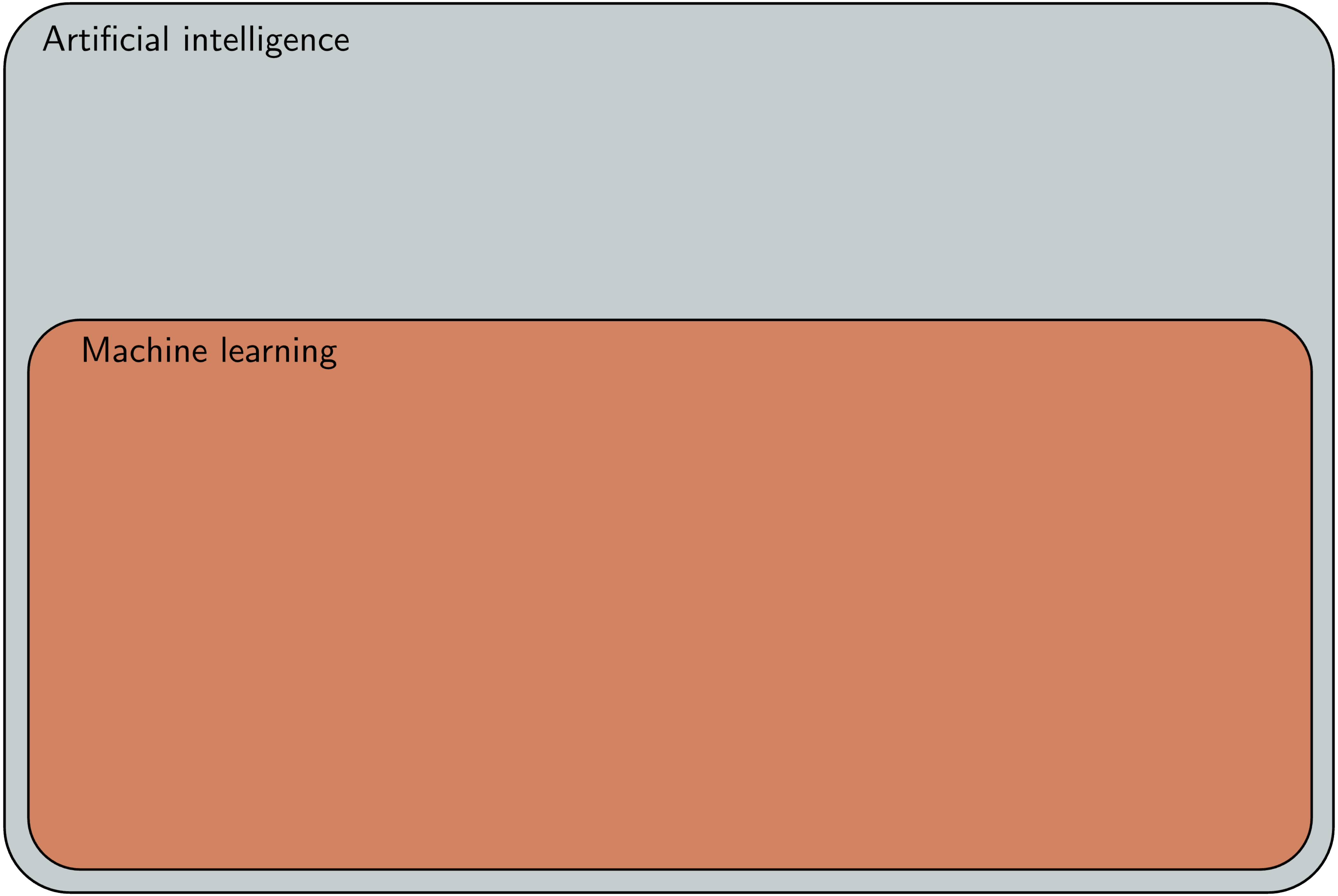
Today

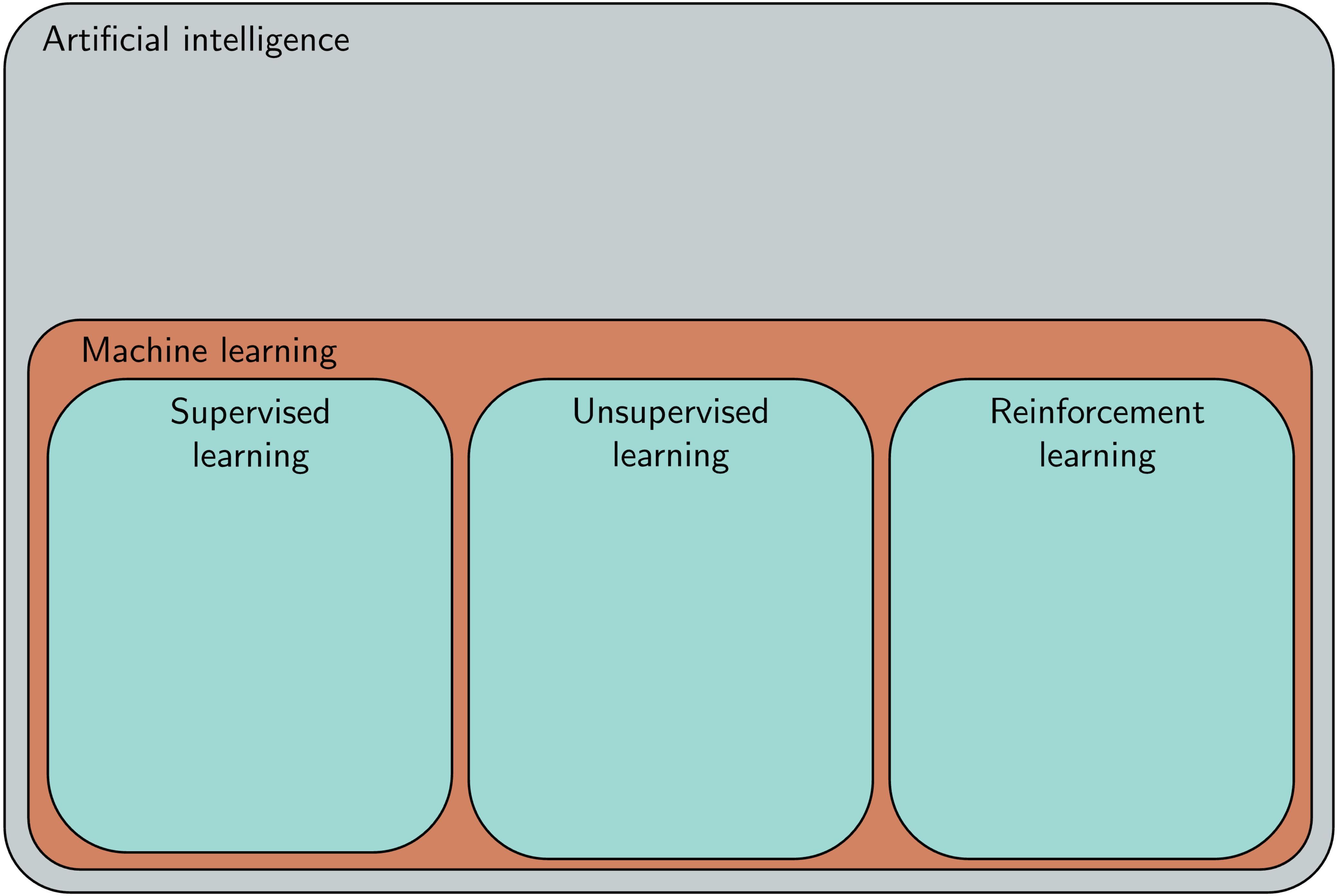
- Introduction to and Applications of Deep Learning
- History of Neural Networks
- Course Logistics

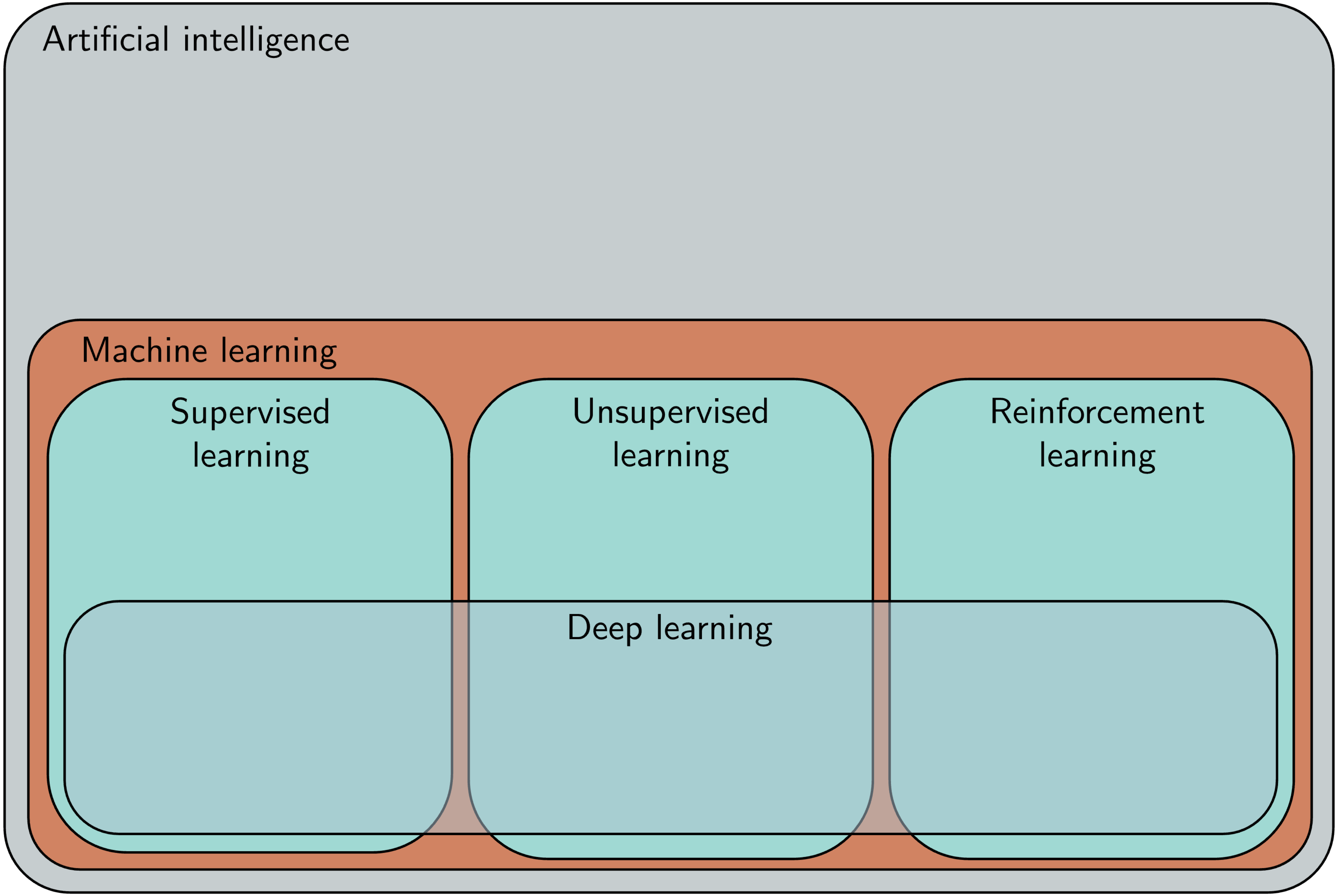
Introduction

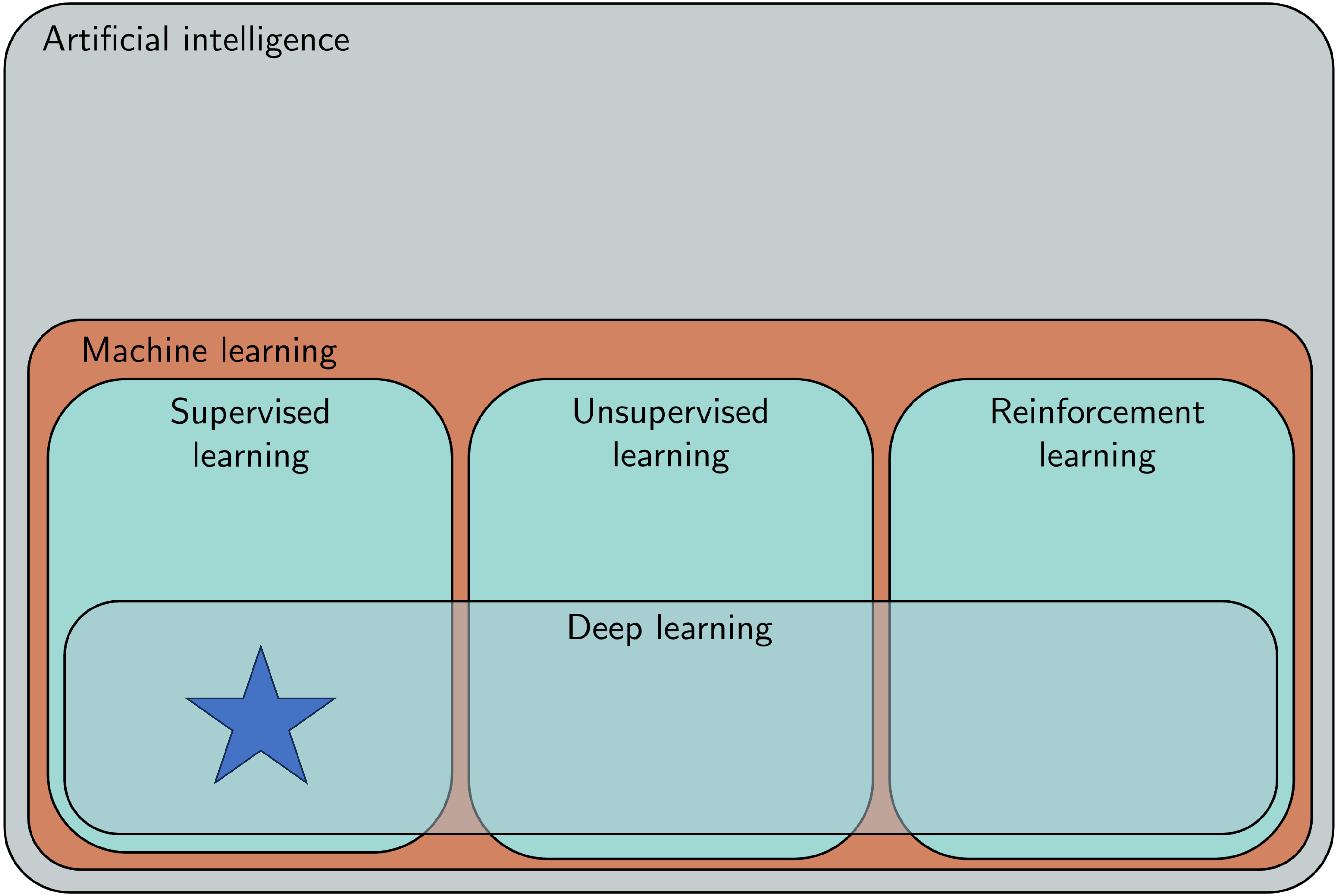
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Artificial intelligence


Artificial intelligence
Machine learning

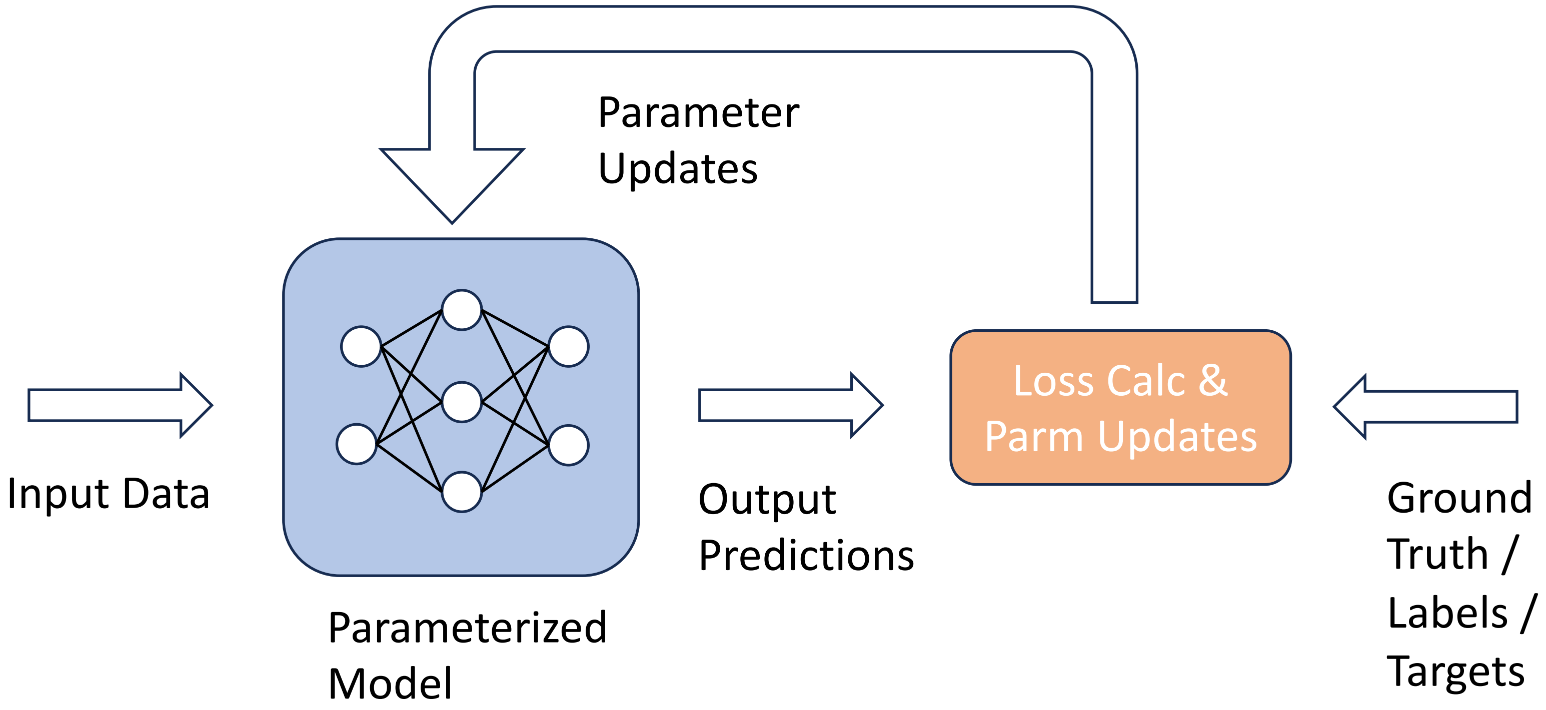

Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

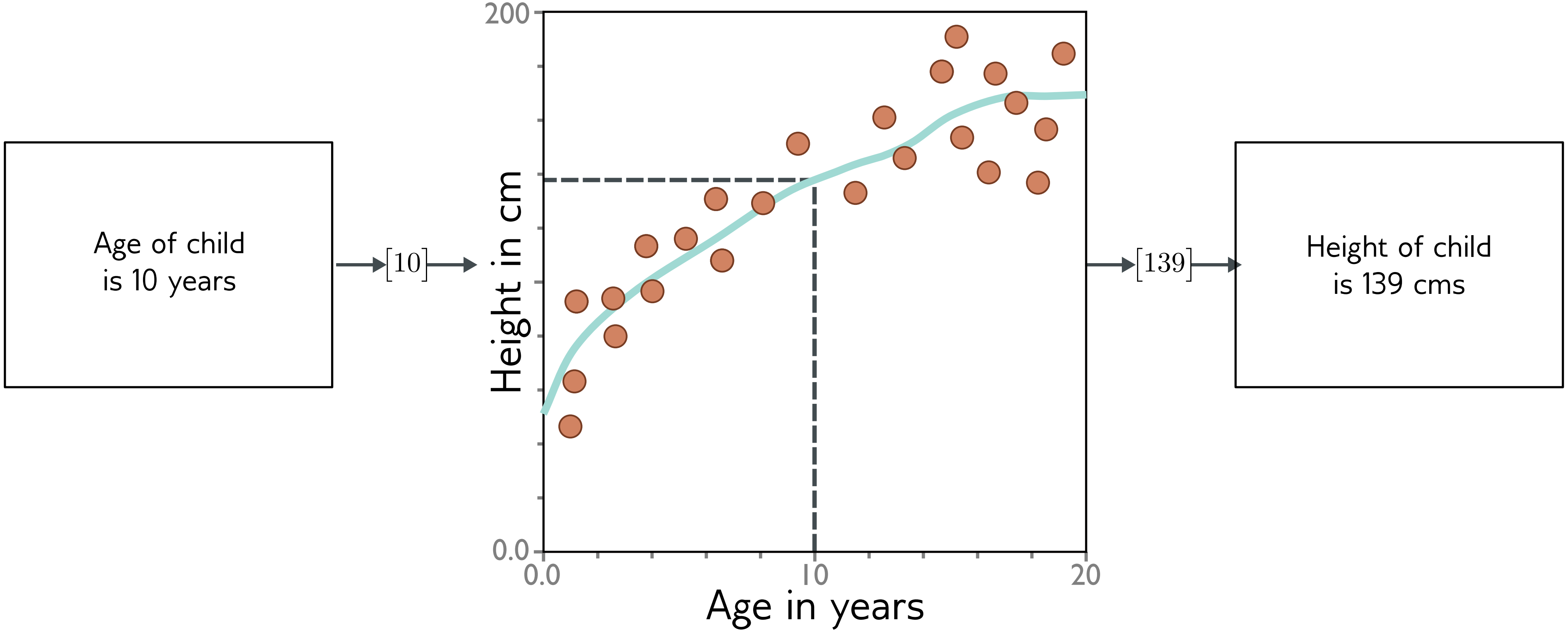

Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Supervised learning

- Define a mapping from input to output
- Learn this mapping from paired input/output data examples

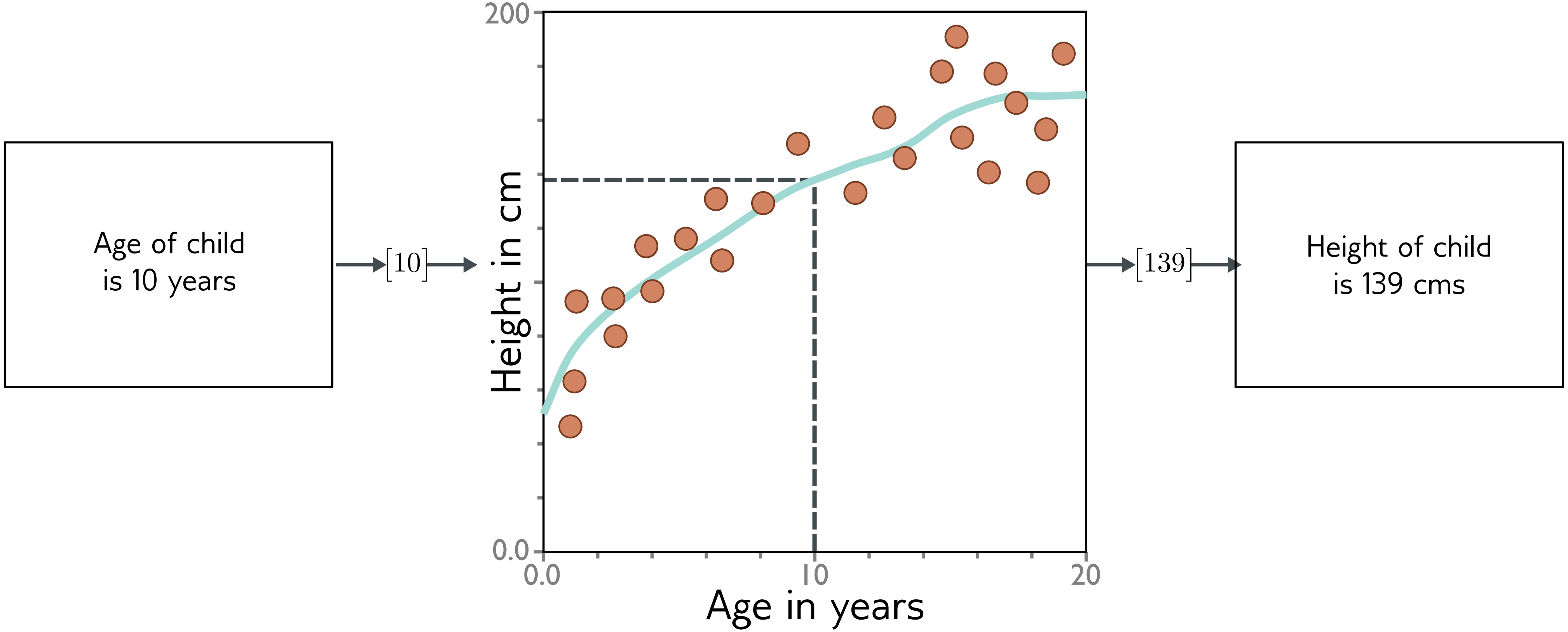


What is a supervised learning model?



- An equation relating input (age) to output (height)
- Search through family of possible equations to find one that fits training data well

What is a supervised learning model?



- Deep neural networks are just a very flexible family of equations
- Fitting deep neural networks = “Deep Learning”

Prediction Types

- Regression
  - Prediction a continuous valued output

- Classification
  - Assigning input to one of a finite number of classes or categories
  - Two classes are a special case

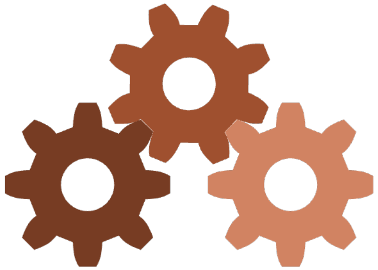
Can be univariate (one output) or multivariate (more than one output)

Regression

Real world input | Model input | Model | Model output | Real world output

6000 square feet,
4 bedrooms,
previously sold for
$235K in 2005,
1 parking spot.

$$\begin{bmatrix} 6000 \\ 4 \\ 235 \\ 2005 \\ 1 \end{bmatrix} \rightarrow$$



Supervised learning model

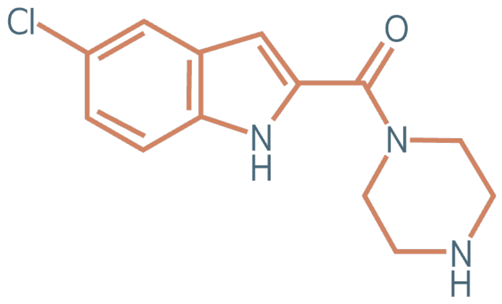
$$\rightarrow [340] \rightarrow$$

Predicted price
is $340k

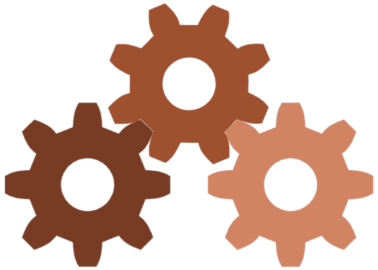
- Univariate regression problem (one output, real value)
- Fully connected network

Graph regression

Real world input | Model input | Model | Model output | Real world output



$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 17 \\ 1 \\ 1 \\ \vdots \end{bmatrix} \rightarrow$$



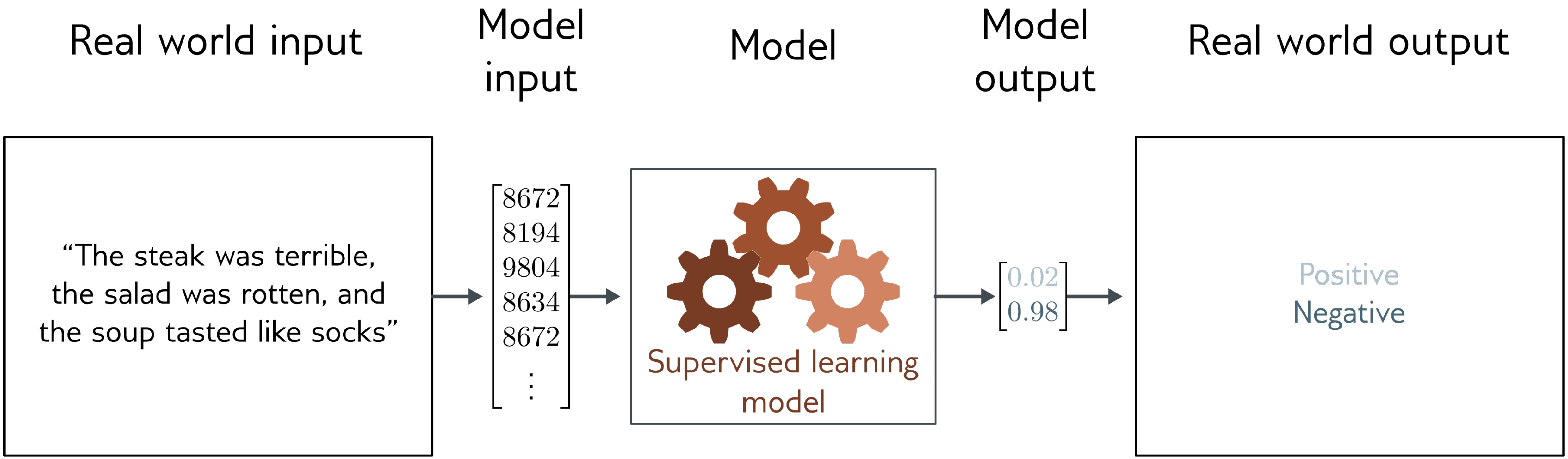
Supervised learning model

$$\rightarrow \begin{bmatrix} -12.9 \\ 56.4 \end{bmatrix} \rightarrow$$

Freezing point is -12.9°C
Boiling point is 56.4°C

- Multivariate regression problem (>1 output, real value)
- Graph neural network

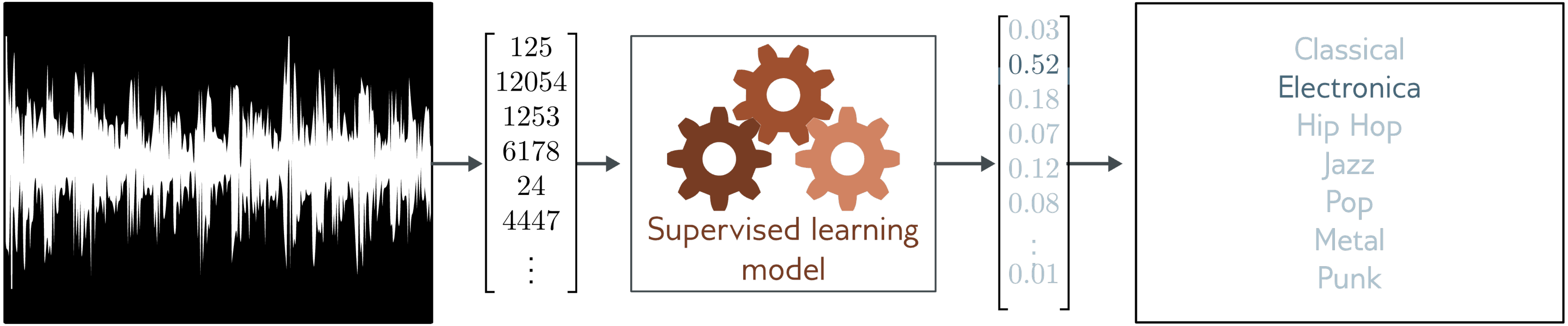
Text classification



- Binary classification problem (two discrete classes)
- Transformer network

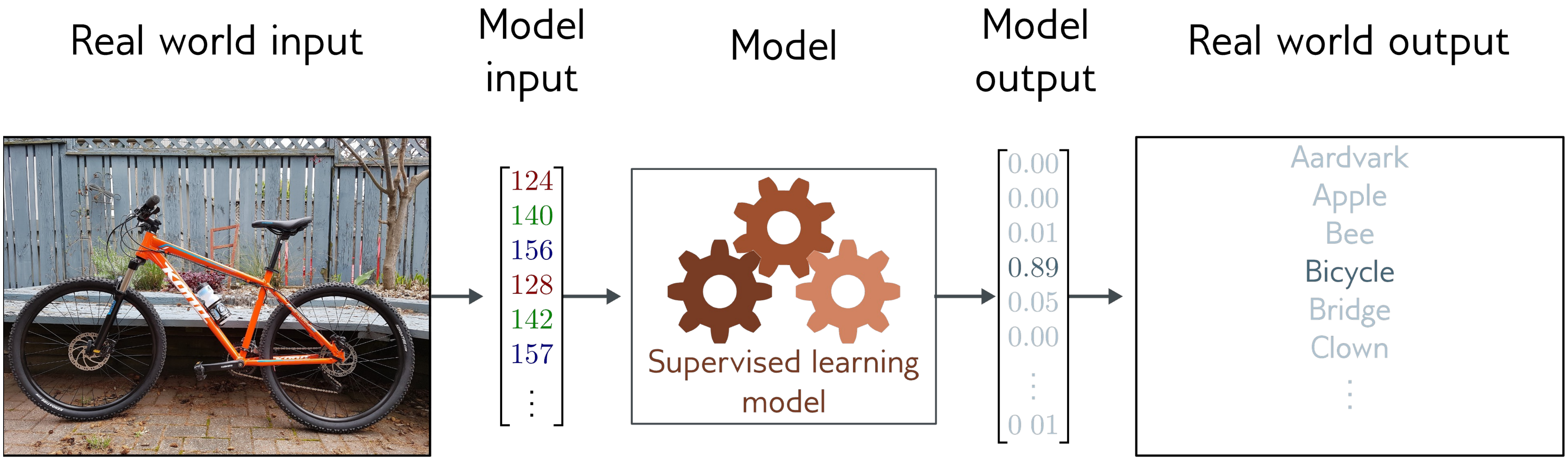
Music genre classification

Real world input | Model input | Model | Model output | Real world output



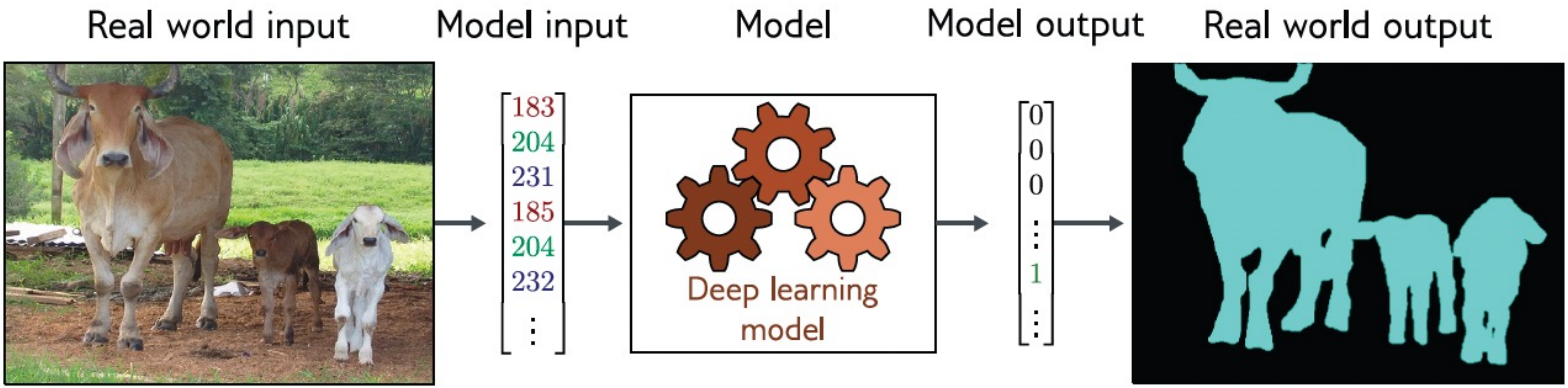
- Multiclass classification problem (discrete classes, >2 possible values)
- Recurrent neural network (RNN)

Image classification



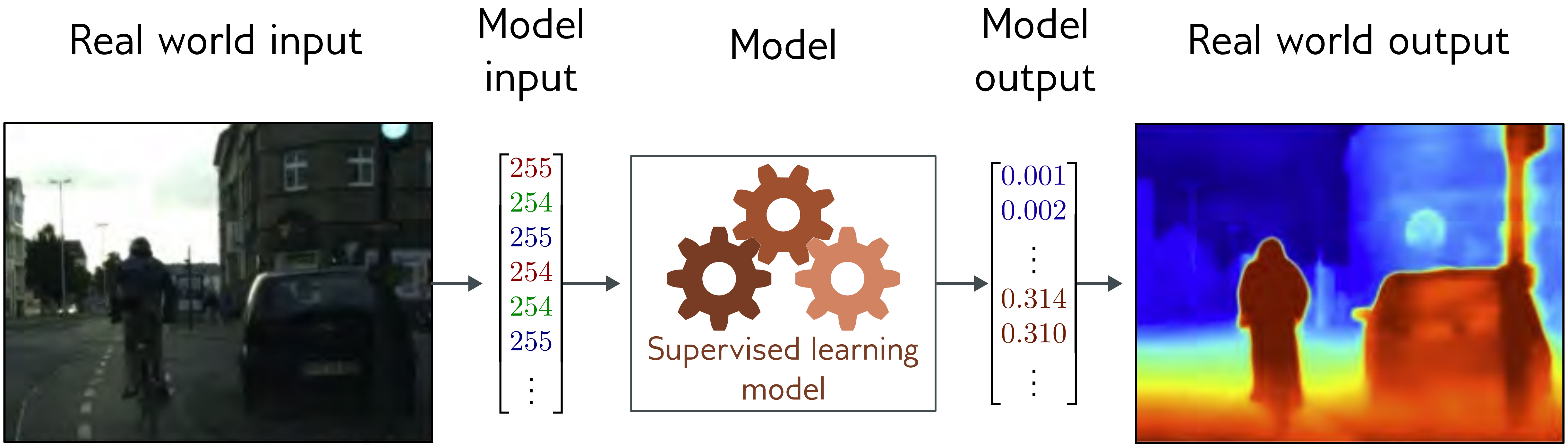
- Multiclass classification problem (discrete classes, >2 possible classes)
- Convolutional network

Image segmentation



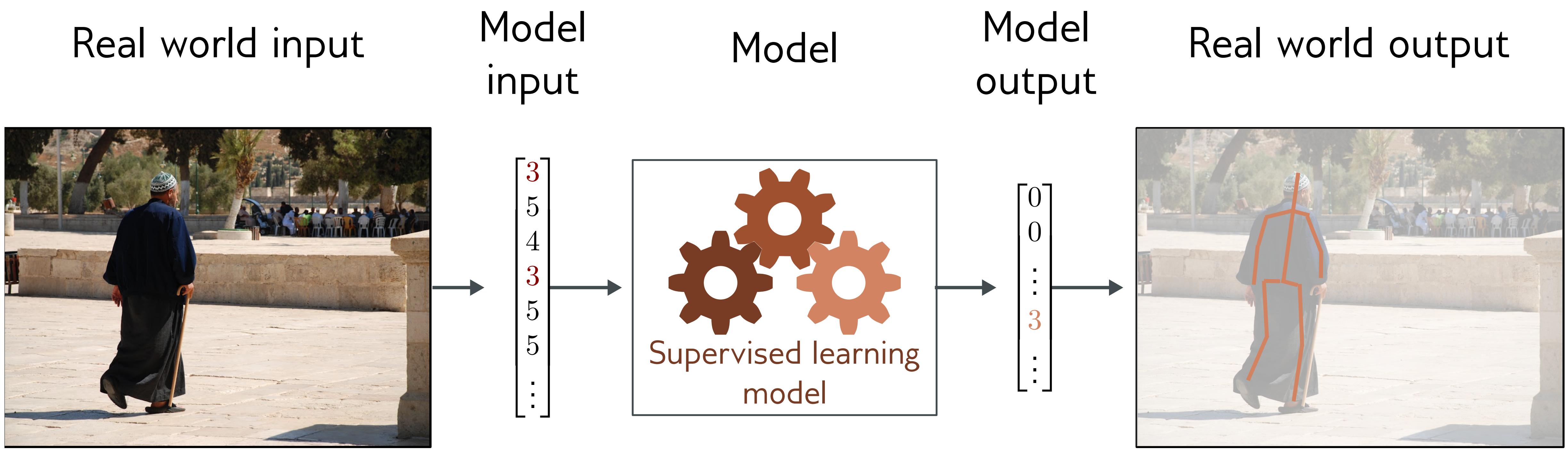
- Multivariate binary classification problem (many outputs, two discrete classes)
- Convolutional encoder-decoder network

Depth estimation



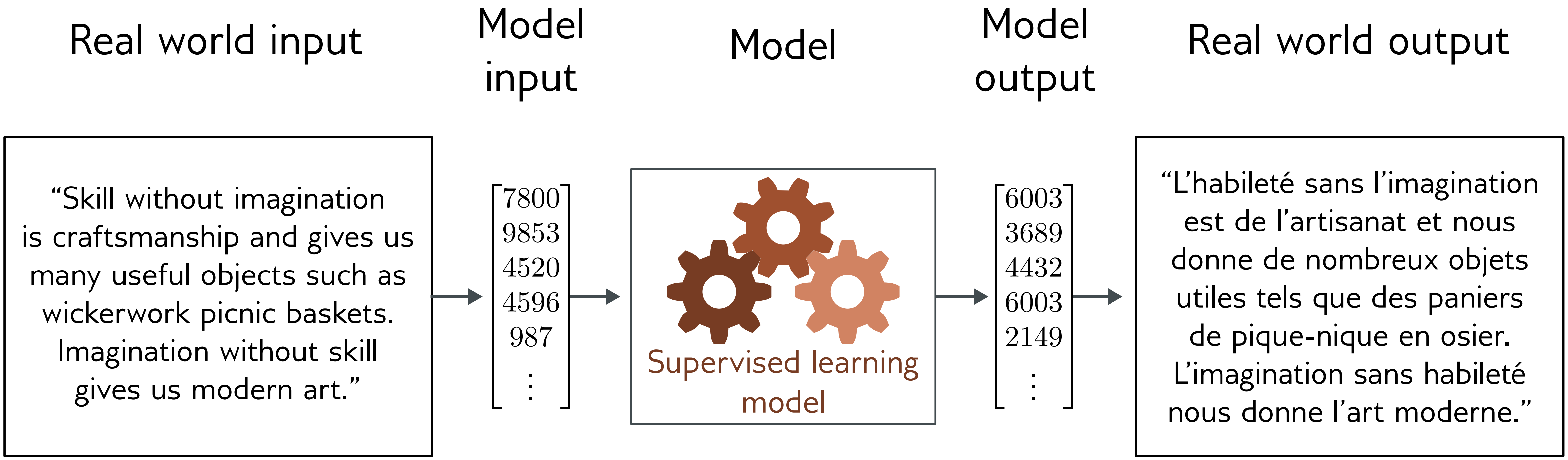
- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Pose estimation



- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Translation

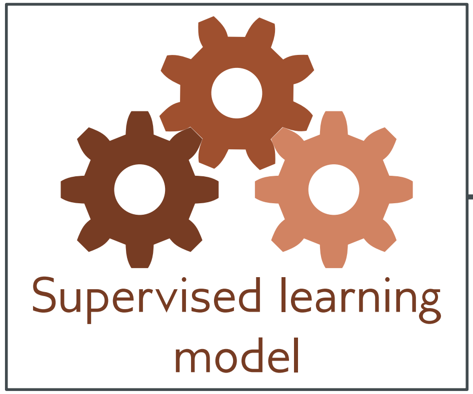


- Encoder-Decoder Transformer Networks

Image captioning

Real world input | Model input | Model | Model output | Real world output

$$\begin{bmatrix} 183 \\ 204 \\ 231 \\ 185 \\ 204 \\ 232 \\ \vdots \end{bmatrix} \rightarrow$$



$$\rightarrow \begin{bmatrix} 1 \\ 5593 \\ 7532 \\ 7924 \\ 1 \\ \vdots \end{bmatrix} \rightarrow$$

"A Kazakh man on a horse holding a bird of prey"

- E.g. CNN-RNN, LSTM, Transformers

Image generation from text

Real world input | Model input | Model | Model output | Real world output

"Teddy bears mixing sparkling chemicals as mad scientists, in a steampunk style."

$$\begin{bmatrix} 8300 \\ 532 \\ 7676 \\ 7898 \\ 883 \\ \vdots \end{bmatrix}$$

Supervised learning model

$$\begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ \vdots \end{bmatrix}$$

What do these examples have in common?

- Very complex relationship between input and output
- Sometimes may be many possible valid answers
- But outputs (and sometimes inputs) obey rules

"A Kazakh man on a horse holding a bird of prey"

Language obeys grammatical rules

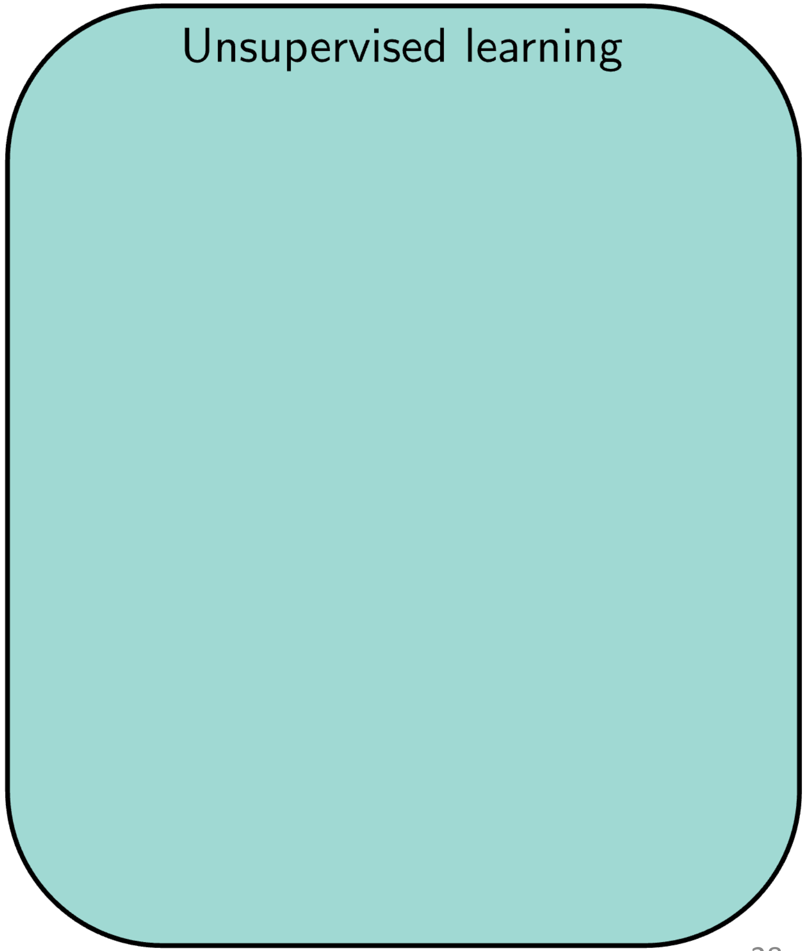
Natural images also have "rules"

Idea

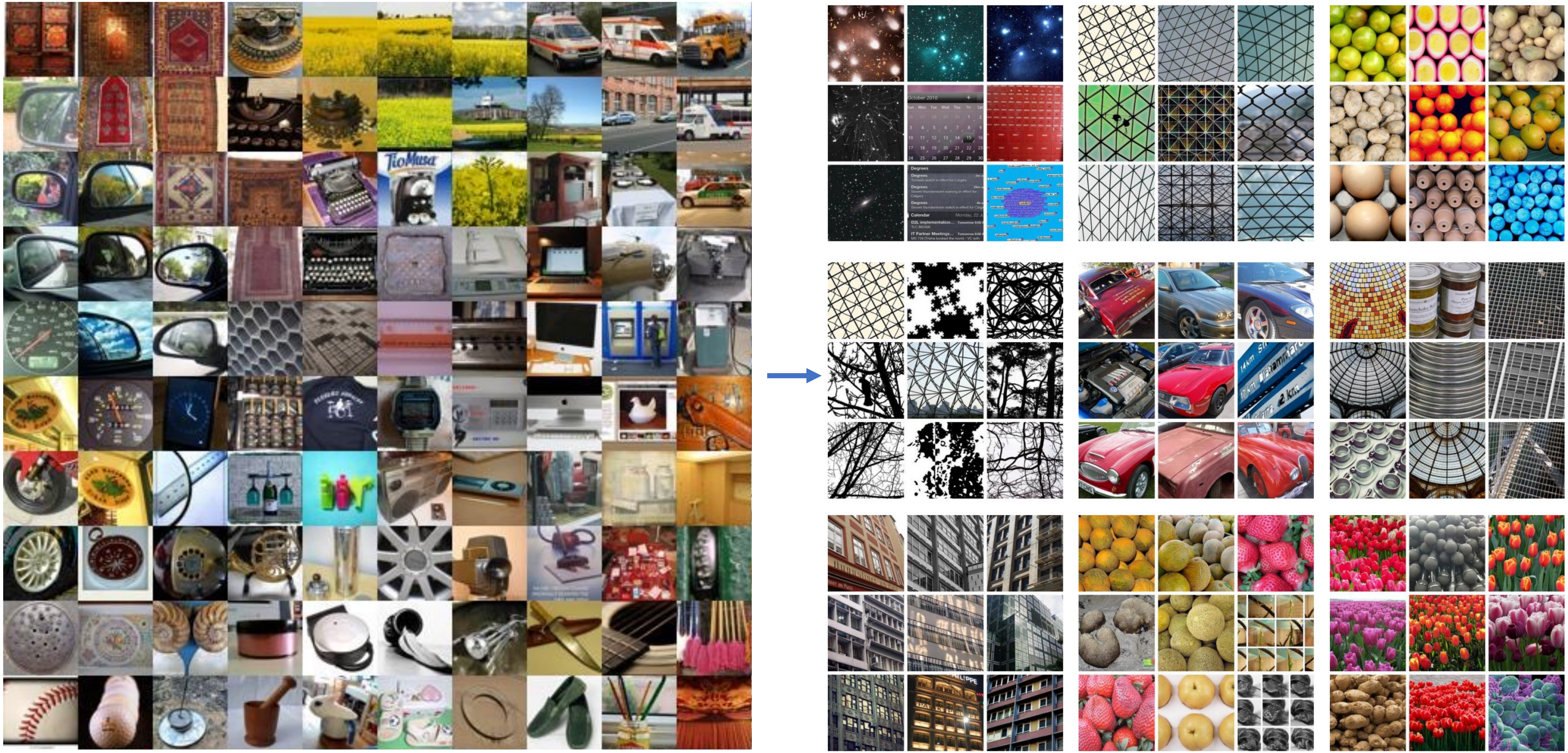
- Learn the “grammar” of the data from unlabeled examples
- Can use a gargantuan amount of data to do this (as unlabeled)
- Make the supervised learning task easier by having a lot of knowledge of possible outputs

Unsupervised Learning

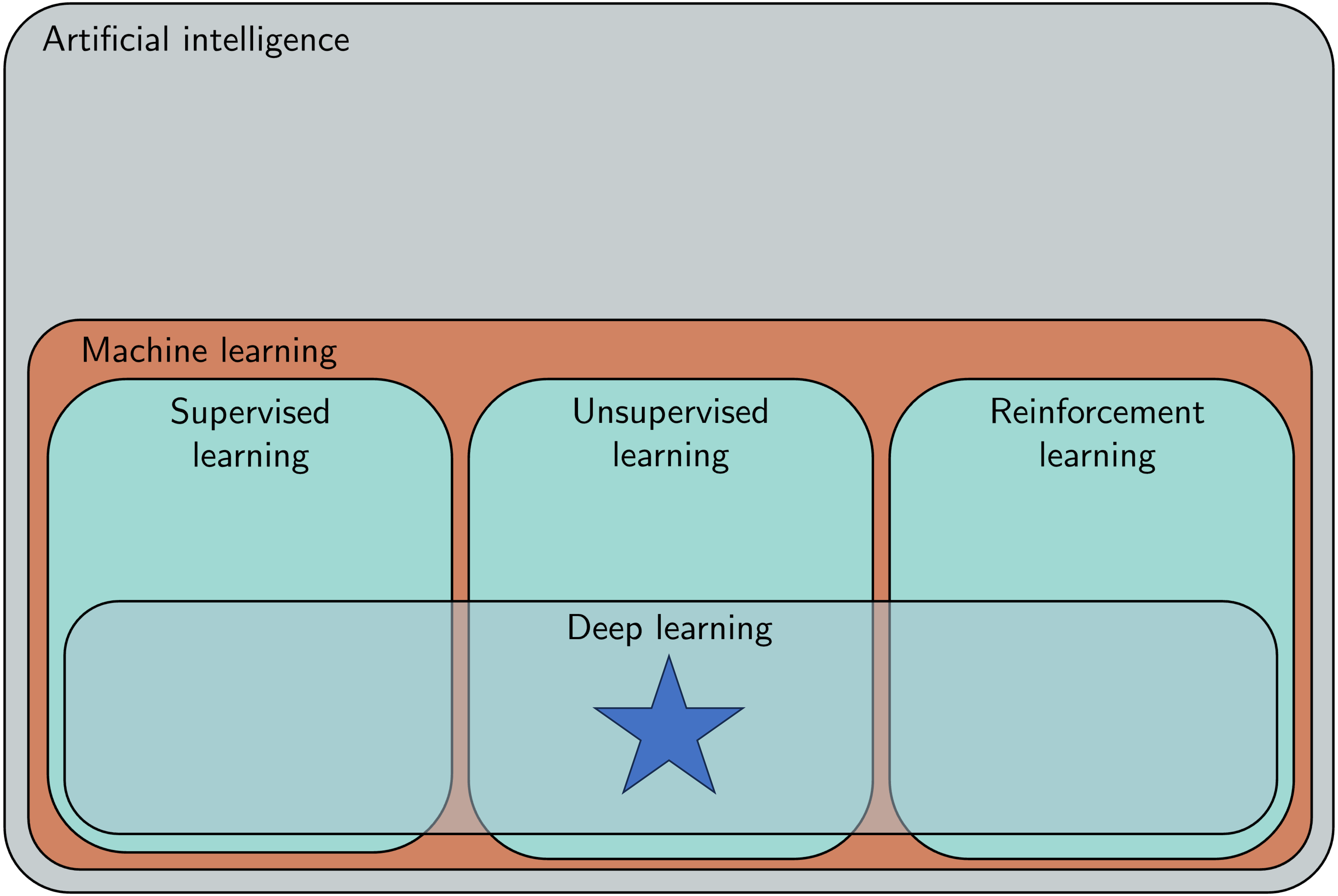
- Learning about a dataset without labels
  - Clustering
  - Finding outliers
  - Generating new examples
  - Filling in missing data



DeepCluster: Deep Clustering for Unsupervised Learning of Visual Features (Caron et al., 2018)

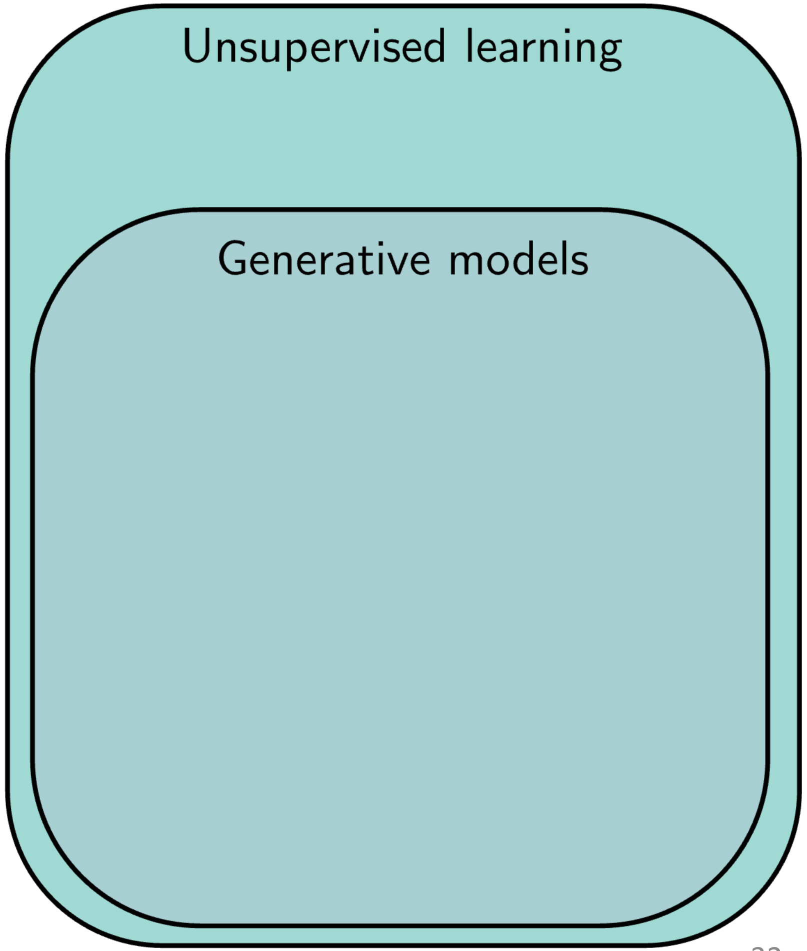


DeepCluster: Deep Clustering for Unsupervised Learning of Visual Features (Caron et al., 2018)


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

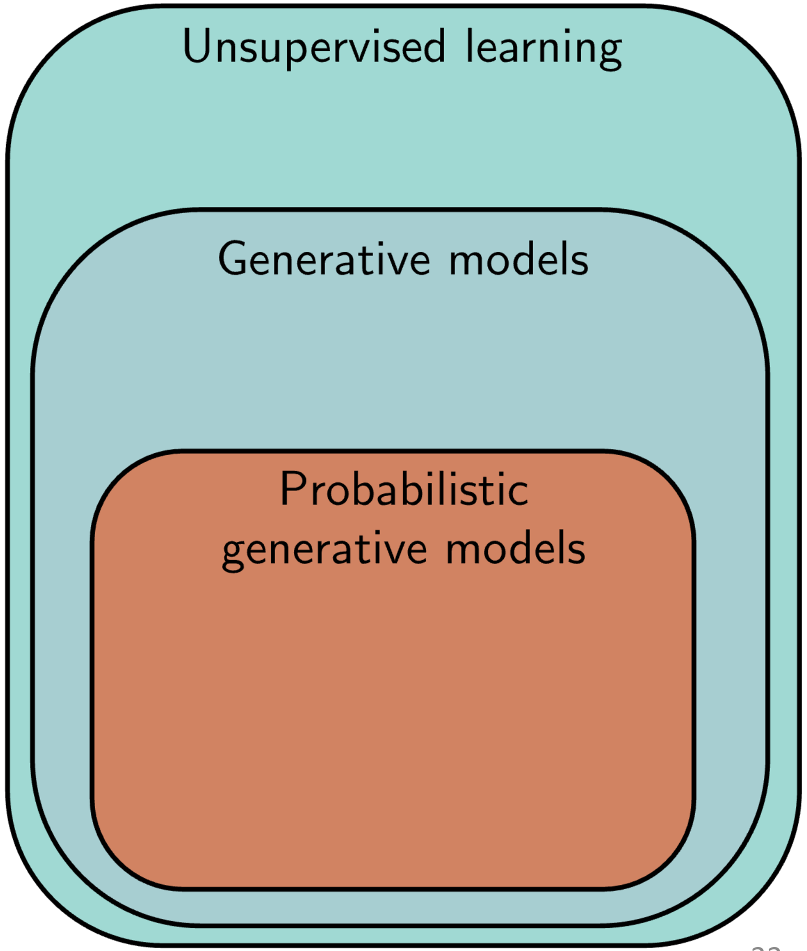
Unsupervised Learning

- Learning about a dataset without labels
  - e.g., clustering
- Generative models can create examples
  - e.g., generative adversarial networks

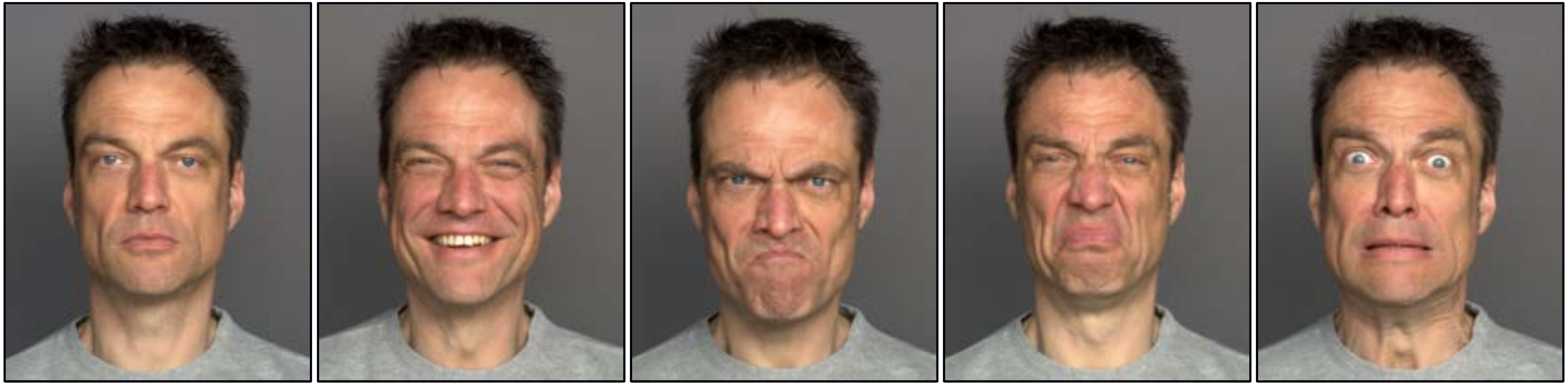


Unsupervised Learning

- Learning about a dataset without labels
  - e.g., clustering
- Generative models can create examples
  - e.g., generative adversarial networks
- Probabilistic Generative Models learn distribution over data
  - e.g., variational autoencoders,
  - e.g., normalizing flows,
  - e.g., diffusion models



Why should this work?



- 42 muscles control all possible expressions
- Restrictions on how faces and heads look subject to physics of illumination and reflectance, etc.

- The “manifold” of possible faces is much, much smaller than the combinatoric collection of pixel values

C. A. C. Holland, N. C. Ebner, T. Lin, and G. R. Samanez-Larkin, “Emotion identification across adulthood using the Dynamic FACES database of emotional expressions in younger, middle aged, and older adults,” *Cognition and Emotion*, vol. 33, no. 2, pp. 245–257, Feb. 2019, doi: 10.1080/02699931.2018.1445981.

Interpolation

Axel Sauer, Katja Schwarz, and Andreas Geiger. 2022. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. In ACM SIGGRAPH 2022 Conference Proceedings (SIGGRAPH '22). Association for Computing Machinery, New York, NY, USA, Article 49, 1–10. https://doi.org/10.1145/3528233.3530738

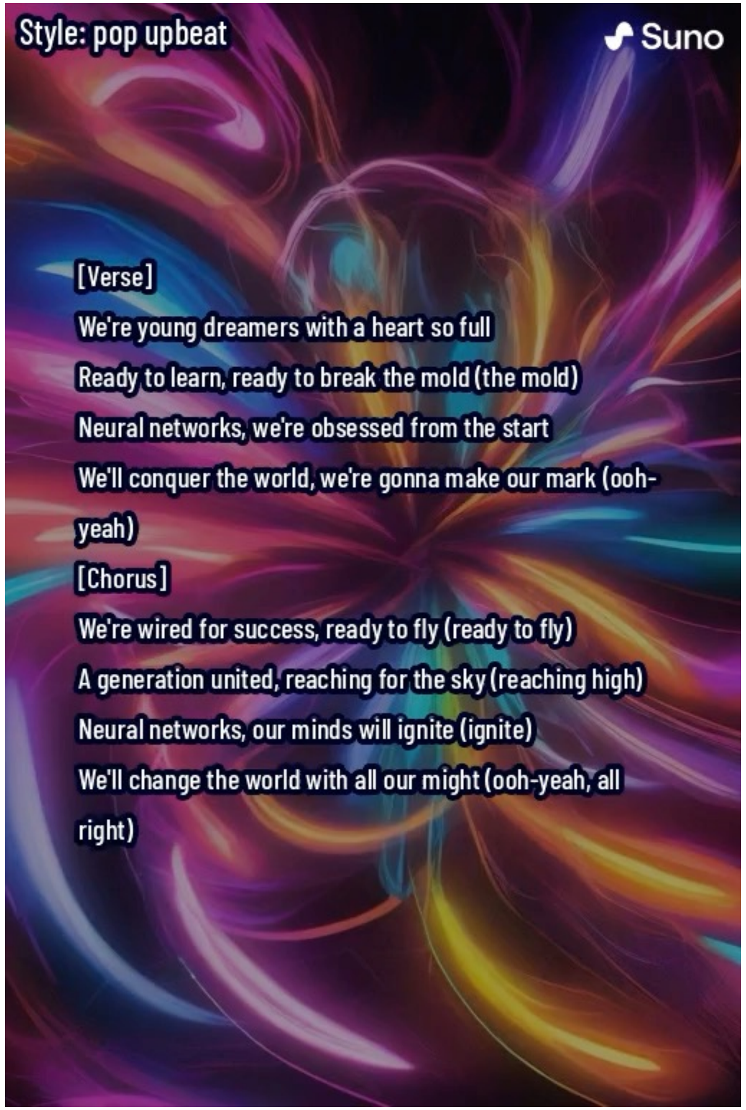
Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with CLIP Latents. arXiv:2204.06125

Conditional synthesis

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., & Norouzi, M. (2022a). Palette: Image-to-image diffusion models. ACM SIGGRAPH, (link)

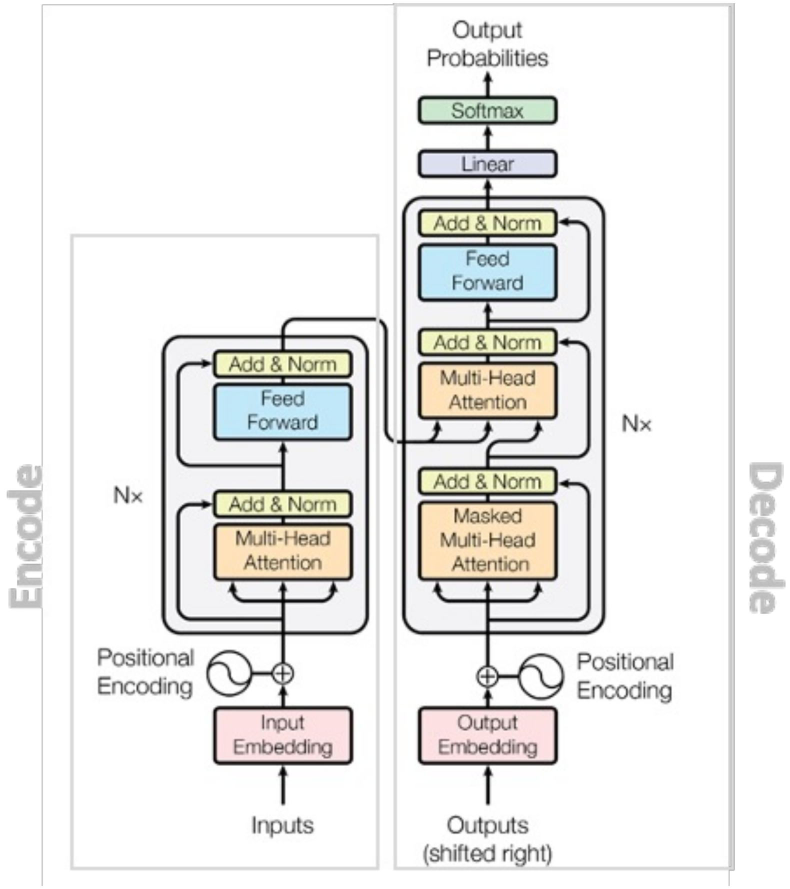
Image/Video/Music Generation

A teenage superhero fighting crime in an urban setting shown in the style of claymation.

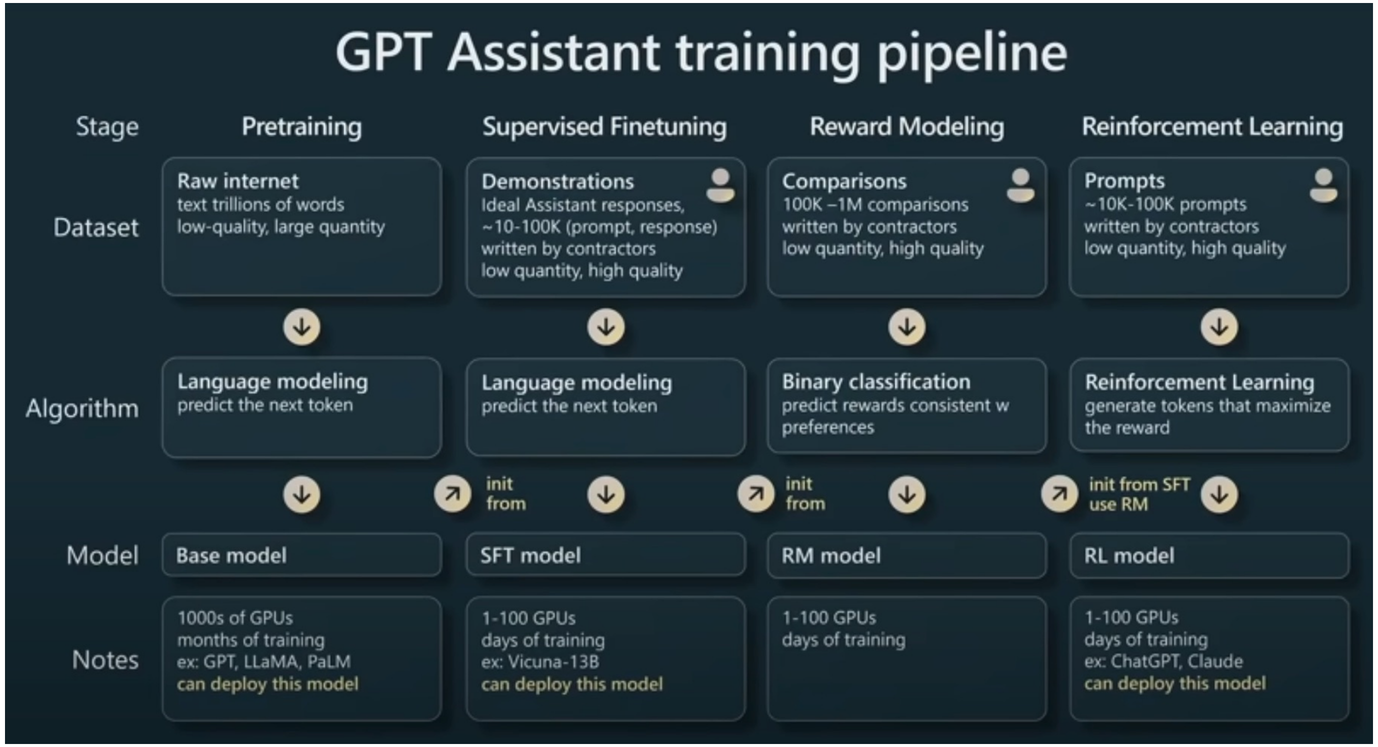


Write a short pop song about students wanting to learn about neural networks and do great things with them.

Transformers, GPTs and Assistants

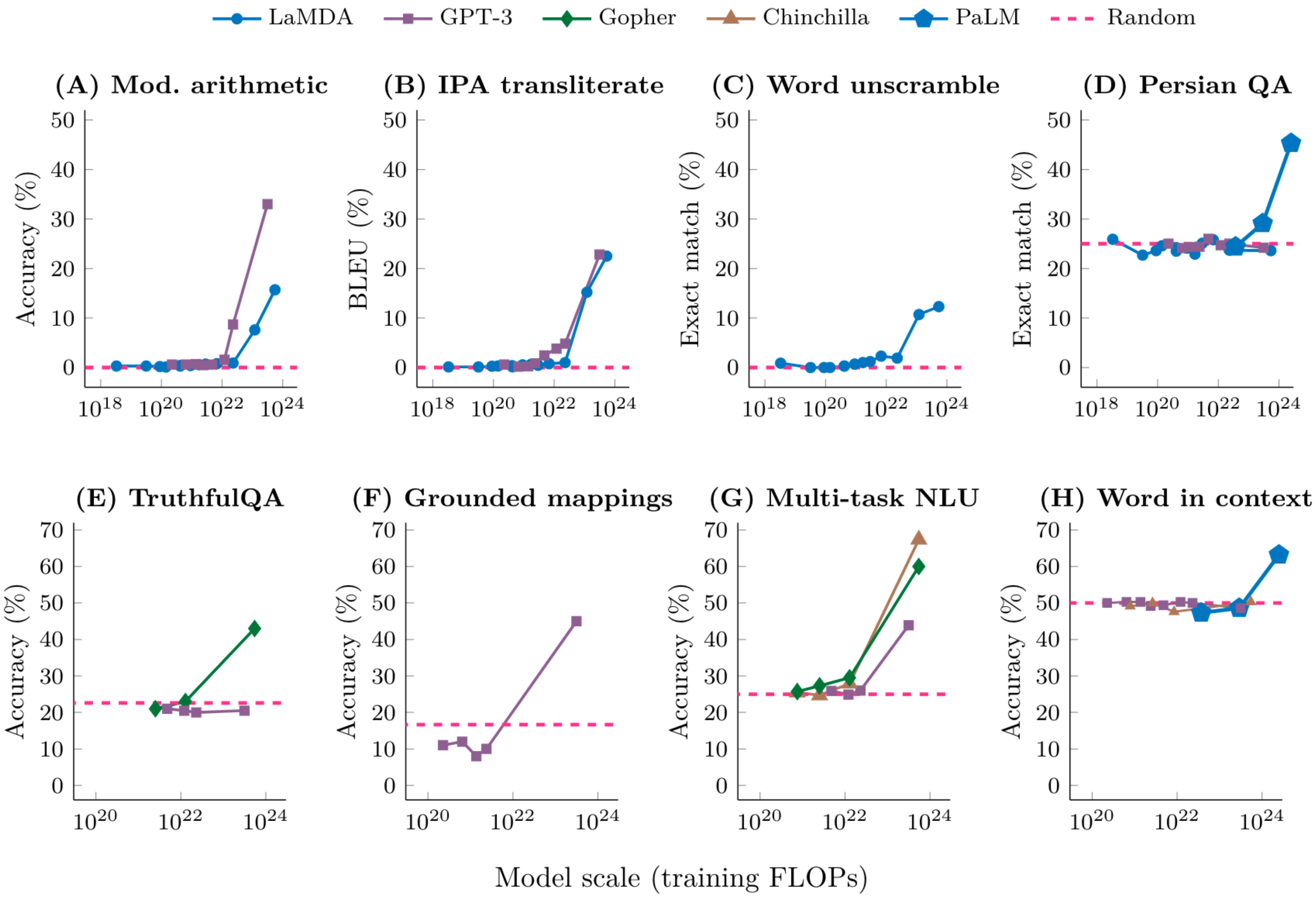


A. Vaswani *et al.*, "Attention is All you Need," presented at the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017, p. 11. [Online]. Available: https://arxiv.org/abs/1706.03762

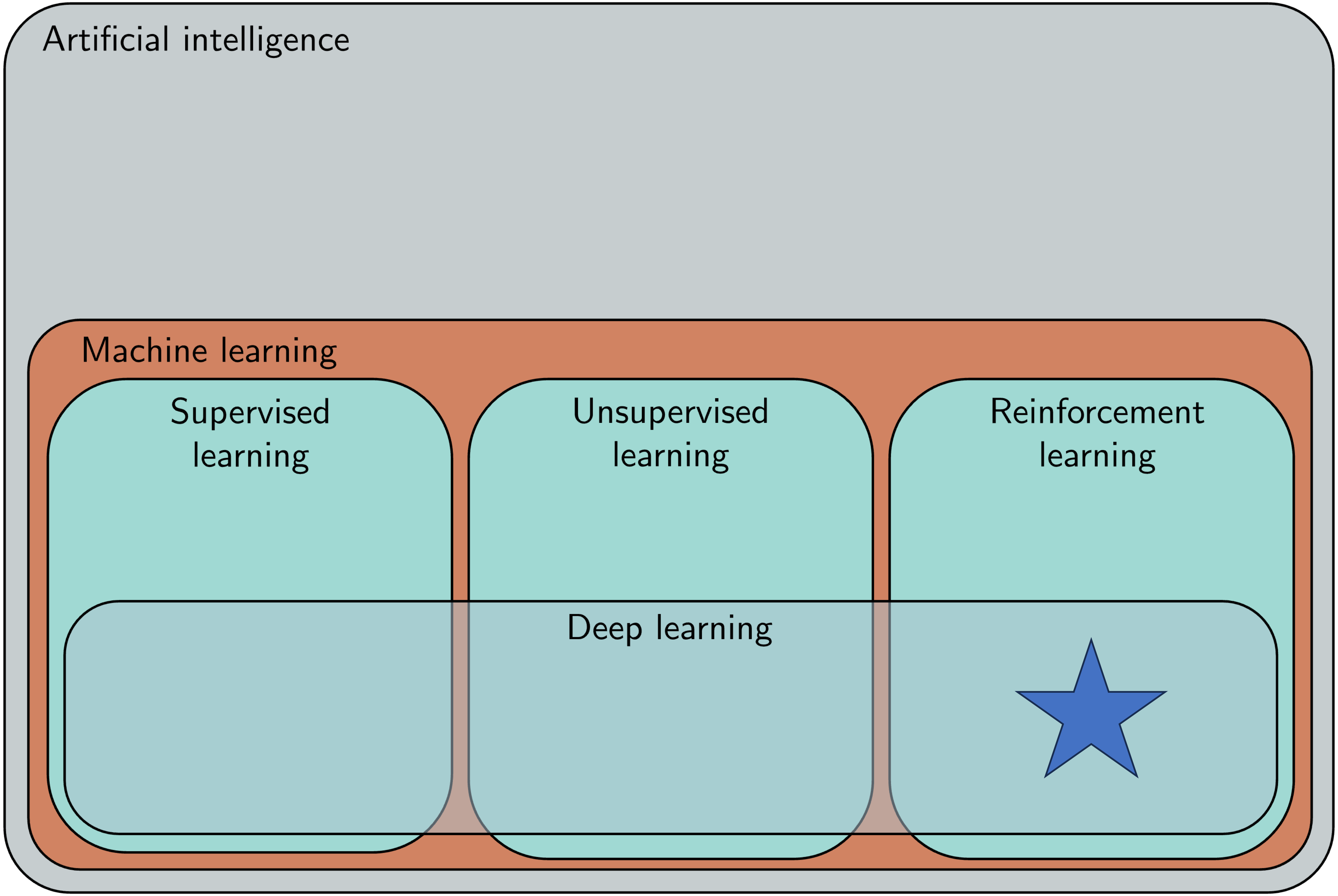


State of GPT, Andrej Karpathy, MS Build Keynote

Emergent Abilities of Large Language Models



“Emergent Abilities of Large Language Models.”(https://arxiv.org/abs/2206.07682) J. Wei et al., Oct. 26, 2022.


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Reinforcement learning

- A set of states
- A set of actions
- A set of rewards

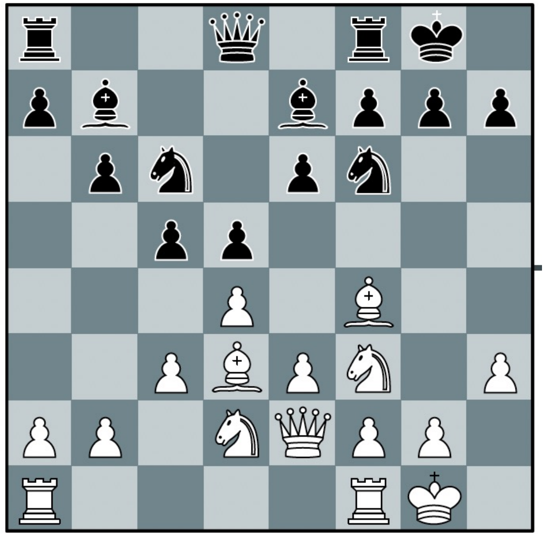
- Goal: take actions to change the state so that you receive rewards

- You don't receive any data – you have to explore the environment yourself to gather data as you go

Example: chess

- States are valid states of the chess board
- Actions at a given time are valid possible moves
- Positive rewards for taking pieces, negative rewards for losing them

State

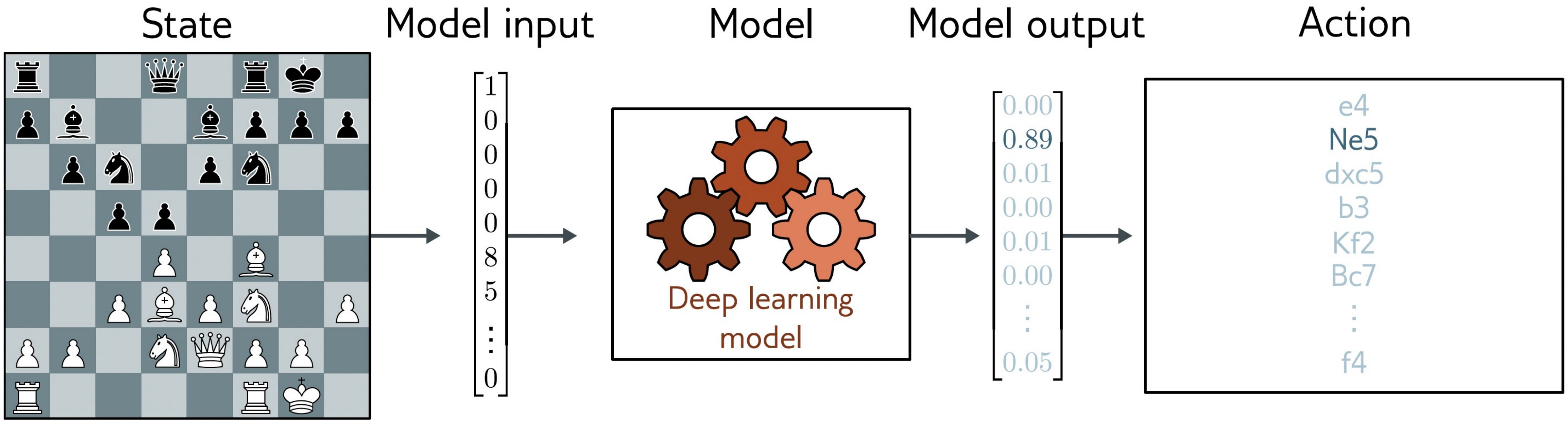


Action

e4
Ne5
dxc5
b3
Kf2
Bc7
⋮
f4

Example: chess

- States are valid states of the chess board
- Actions at a given time are valid possible moves
- Positive rewards for taking pieces, negative rewards for losing them



Why is this difficult?

- Stochastic
  - Make the same move twice, the opponent might not do the same thing
  - Rewards also stochastic (opponent does or doesn’t take your piece)
- Temporal credit assignment problem
  - Did we get the reward because of this move? Or because we made good tactical decisions somewhere in the past?
- Exploration-exploitation trade-off
  - If we found a good opening, should we use this?
  - Or should we try other things, hoping for something better?

History of Neural Networks

Abbreviated History of NNs

- 1943: McCulloch & Pitts – Calculus of neurons
- 1947-49: Donald Hebb – Plasticity of neurons
- 1956: Minsky, McCarthy, Shannon… Dartmouth Summer Research Project on AI
- 1957: Rosenblatt – Perceptron, HW implementation of 20x20 CV
- 1959: Hubel & Weisel – Visual cortex and receptive fields
- 1960: Widrow & Hoff – Adaptive Linear Neuron (ADALINE)
- 1969: Minsky & Papert – Perceptrons: computation limitations of neurons

Continued (abbreviated) History

- 1979 – Fukushima: Neocognitron, cascade of neural structures that can classify shapes, invariant to shift, learned from data
- 1982 – Hopfield Networks, recurrent artificial neural networks
- 1983 – Hinton & Sejnowski: Boltzmann Machines
- 1985 – Rumelhart, Hinton, Williams: Practical backpropagation
- 1989 – LeCun – Backprop on Convolutional Neural Networks
- 1991 – Bottou & Galinari – Automatic differentiation (autograd)
- 2012 – AlexNet (DNN on GPU trained on ImageNet)
- 2016 – Kaiming He: ResNet

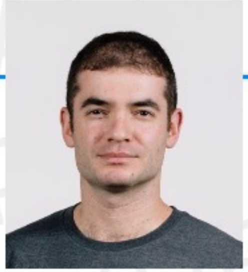
A Brief History of Transformers

2000

Yoshua Bengio*

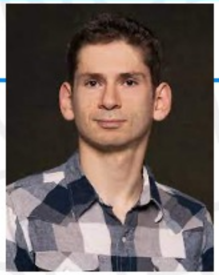
2014

Ilya Sutskever*



2014

Dzmitry Bahdanau*



2017

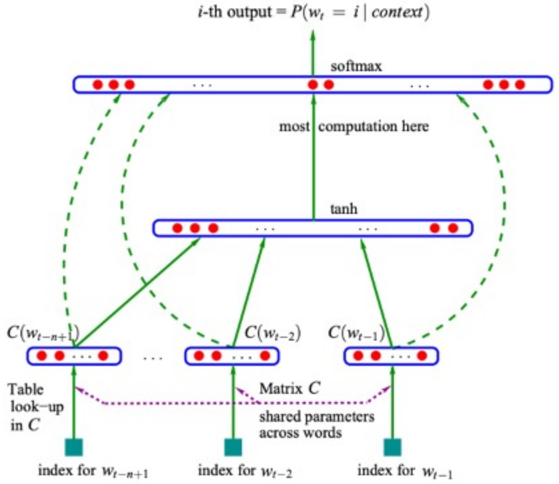
A Team at Google

Use LSTMs

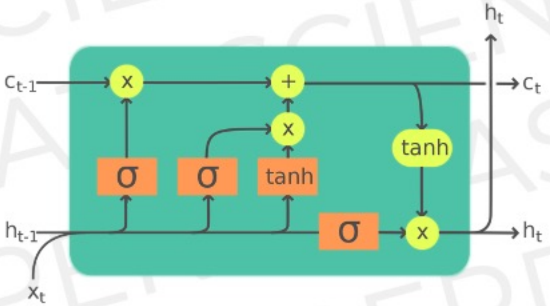
Add Attention

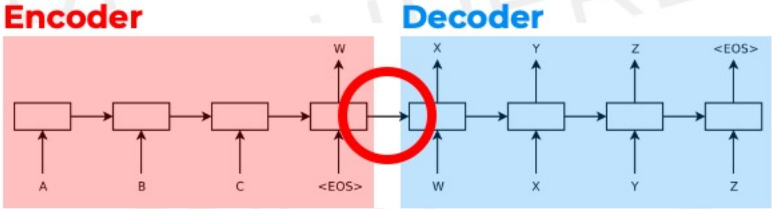
Remove LSTMs

A Neural Probabilistic Language Model

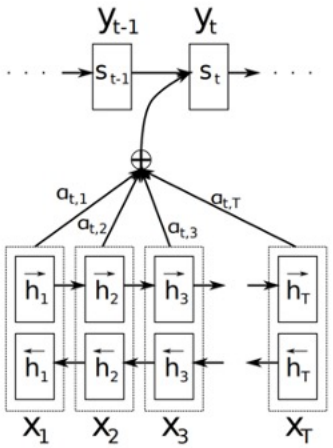


Seq-to-Seq Learning with Neural Networks

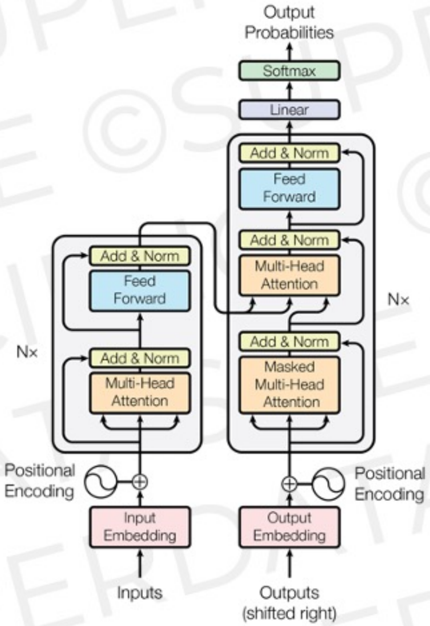




Neural Machine Translation by Jointly Learning to Align and Translate



Attention is all you need



*And others; Chronological analysis inspired by Andrej Karpathy's lecture, youtube.com/watch?v=XfpMkf4rD6E

Course Logistics

Course Website: https://dl4ds.github.io/sp2024/

BOSTON UNIVERSITY

Boston University Faculty for Computing and Data Science

Deep Learning for Data Science (DL4DS)

Spring 2024

HOME SCHEDULE LECTURES ASSIGNMENTS PROJECT MATERIALS

Deep Learning for Data Science (DL4DS) / Spring 2024

Announcements

- Jan 10, 2024:
  Added lecture titles and first version of the project description. The notes, codes and slides links are still placeholders.
- Dec 4, 2023:
  This course web site is under active construction. Check back regularly for updates.
  The Schedule, Lectures, Assignments, Projects and Materials pages are being updated and will be posted soon.

Course Abstract

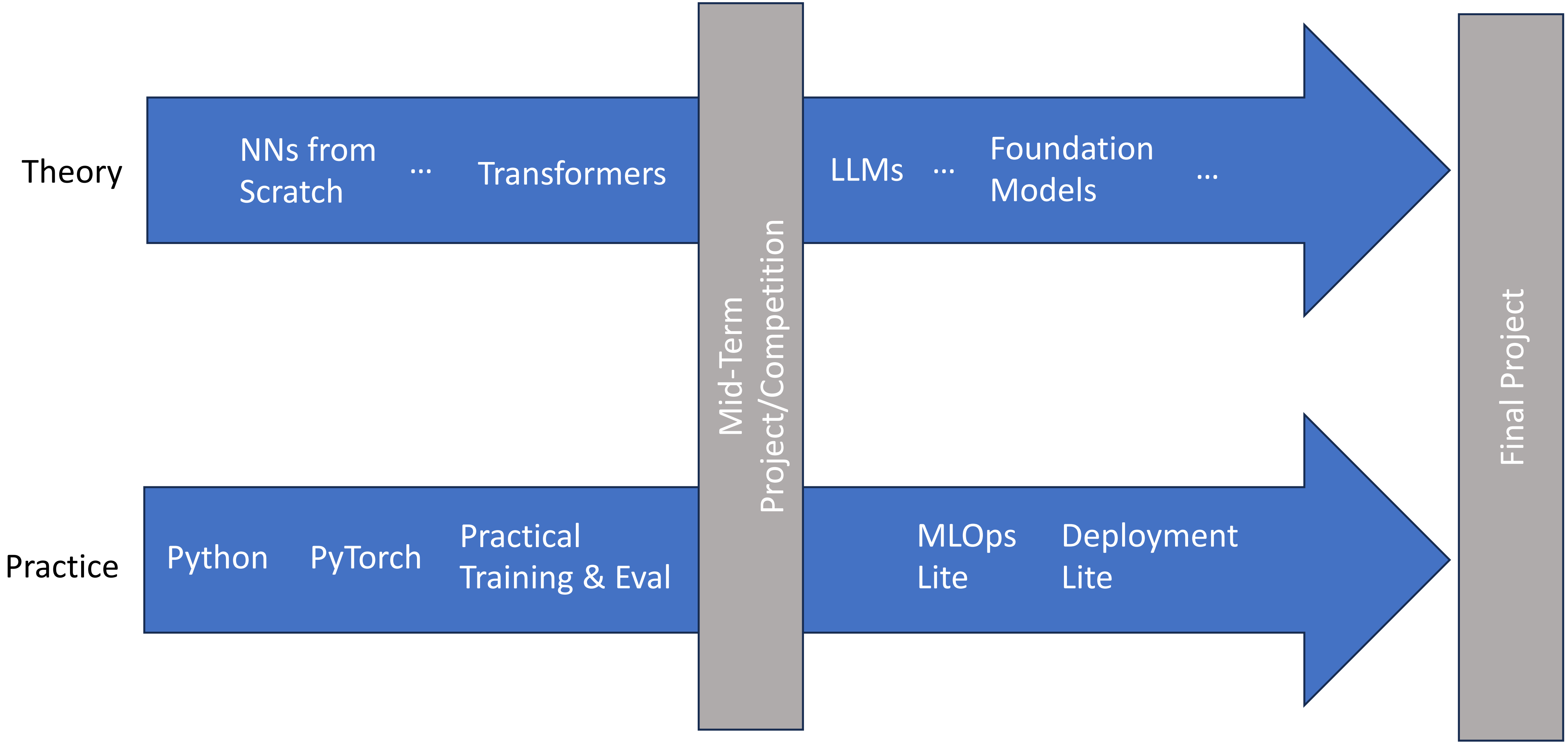
In this course we will dive into Deep Learning. We'll balance important theoretical concepts with hands on network training and applications using modern deep learning python frameworks. We'll explore numerous network architectures like CNNs, transformers, and the rapidly developing state-of-the-art of large pre-trained foundation models. You'll have the chance to apply what you've learned in a final project.

Lectures: Tuesdays and Thursdays, 3:30pm – 4:15pm
Location: CAS 208

Discussion Session I: Wednesdays, 11:15am – 12:05pm
Location: CDS 164

Discussion Session II: Wednesdays, 3:35pm – 4:25pm
Location: CDS 1526

Balancing Theory and Practice – Two Tracks



Subject to some tweaks

Course Outline -- Lectures

First Half

1. Intro, Project Ideas, Course Logistics
2. Supervised Learning
3. Shallow Networks
4. Deep Networks
5. Loss Functions
6. Fitting Models
7. Gradients
8. Initialization
9. Measuring Performance
10. Regularization
11. Convolutional Neural Networks
12. Residual Networks
13. Transformers
14. **Mid-term Project Presentations**
15. **Mid-term Project Presentations**

Python & PyTorch Practice

Spring Break

Second Half

16. Language Embeddings and Models
17. Foundation Models
18. Fine Tuning (LoRA, ...) Transfer Learning, etc.
19. Cognitive Architecture, RAG, Chatbots
20. Multimodal Transformers and Foundation Models
21. Graph Neural Networks
22. Unsupervised Learning
23. GANs
24. Diffusion Models
25. Reinforcement Learning
26. _Tentative:_ Future Directions
27. Final Project Presentations
28. Final Project Presentations

Python & PyTorch Applications

Discussions Outline – Python and PyTorch

Subject to some tweaks

First Half

Week

1. Intro to Pytorch, Tensors, and Tensor Operations.
2. Derivatives, Autograd, and Computational Graphs in Pytorch.
3. Workflow of training a custom model using torch.nn. Loss functions and activation functions.
4. Deep-dive 1: How to read data, how to load data, how to preprocess data in Pytorch (creating a custom dataset, how to use collate_fn in a dataloader, tokenizing text, image augmentation etc).
5. Deep-dive 2: Looking deeper into the available building blocks in torch.nn.
6. Deep-dive 3: Measuring Model performance, Maintaining Logs during training (logging, weights and biases etc), hyper parameter tuning/search using optuna or other frameworks.
7. Debugging Models, Visualizing intermediate layers, explainability/interpretability. (Trying to open up the black box)

Spring Break

Second Half

Week

8. TBD
9.
10.
11.
12.
13.
14.

Course Project -- https://dl4ds.github.io/sp2024/project/

- Work in individually or in teams of 2-3
- Can be application, algorithmic, theoretical or combination thereof
- Some example ideas on the website, but propose new ones!
- Project proposal due Feb. 16
- Deliverables:
  - Code in GitHub repo
  - Report/paper
  - 3-4 minute video
- More info later, but feel free to brainstorm with me now

Possible Projects

- Class AI Tutor
- Teacher's AI Assistant
- CDS Curriculum AI Assistant
- CDS Building Recycling Advisor
- Media Bias Detection
- Herbaria Foundation Model
- Modern Implementation of Classic Models
- Develop a new dataset for a new class of problem and an initial model
- *…your ideas here…*

Look at Kaggle, Conferences, Workshops, Datasets….

Jupyter Notebooks / NBGrader

- Short Jupyter notebooks to help ground theory with python
- We'll be experimenting with using NBGrader for autograding and manually grading
  - Pay attention to instructions on how to collect and submit your notebooks
- You can do them on Google Colab or in your own environment
- First notebook is out to get you started… reach out with questions

Homework

- Short assignments every week to help you check your understanding

- The first assignment is your Statement of Purpose
  - What do you want to get out of the class?
  - What areas in particular interest you?
  - What’s your learning style?

Mid-term Kaggle Competition

- Work individually
- Details to be posted
- In-class presentations on your approach and results

Grade Weighting – *No “High Stakes” Exams*

Item	Percentage
Final Project	40%
Mid-term Project/Competition	25%
Jupyter Notebooks	15%
Homeworks	15%
Class Participation/Attendance	5%

Generative AI Assistance (GAIA) Policy

https://dl4ds.github.io/sp2024/index.html#gaia-policy

1. Give credit to AI tools whenever used, even if only to generate ideas rather than usable text, illustrations or code.

…

3. When using AI tools on _coding_ assignments, unless prohibited
    1. Add the prompt text and tool used as comments before the generated code. Clarify whether the code was used as is, or modified somewhat, moderately or significantly.

…

5. Use AI tools wisely and intelligently, aiming to deepen understanding of subject matter and to support learning.

Focus on your learning objectives!

How to succeed in this class

- Do the readings before the lecture – come with questions
- Stay on top of the Jupyter notebook and problem set knowledge checks
- Think about project ideas early. Talk about them with peers, advisors and instructors early and often.
- Put the time in on mid-term competition and final project… there's ramp up effort on both and the real returns come towards the end
- Be mindful of generative AI assistance. Your goal is proficiency and fluency. GAIA can rob you of that

Most importantly!!

- Pursue your curiosity
- Challenge yourself intellectually in this exciting and fast-moving area
- Explore your interests

- … and let's have fun!

Feedback?

Link