BOSTON UNIVERSITY

Deep Learning for Data Science DS598 B1

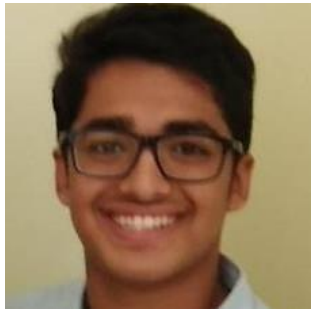
https://dl4ds.github.io/sp2025/

Introduction and Course Overview

Staff

Thomas Gardos
Instructor

tgardos@bu.edu
thomas-gardos
trgardos



Xavier Thomas
Teaching Assistant

xthomas@bu.edu
xavierohan

Hemangi Suthar
Teaching Assistant

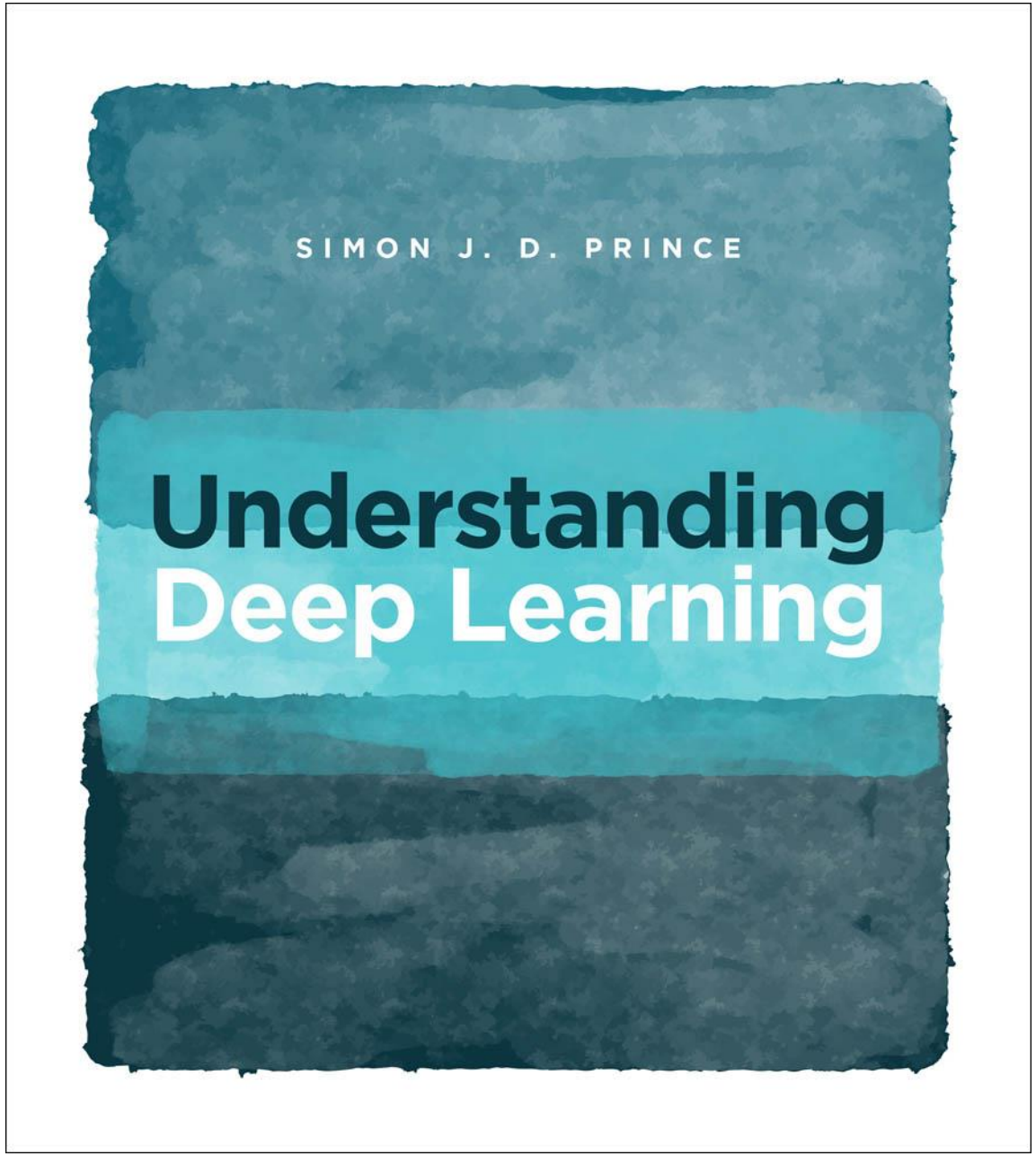
hemangis@bu.edu
Hemangi31

Terrier Tutor
AI Course Assistant V3
(Coming Soon)

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



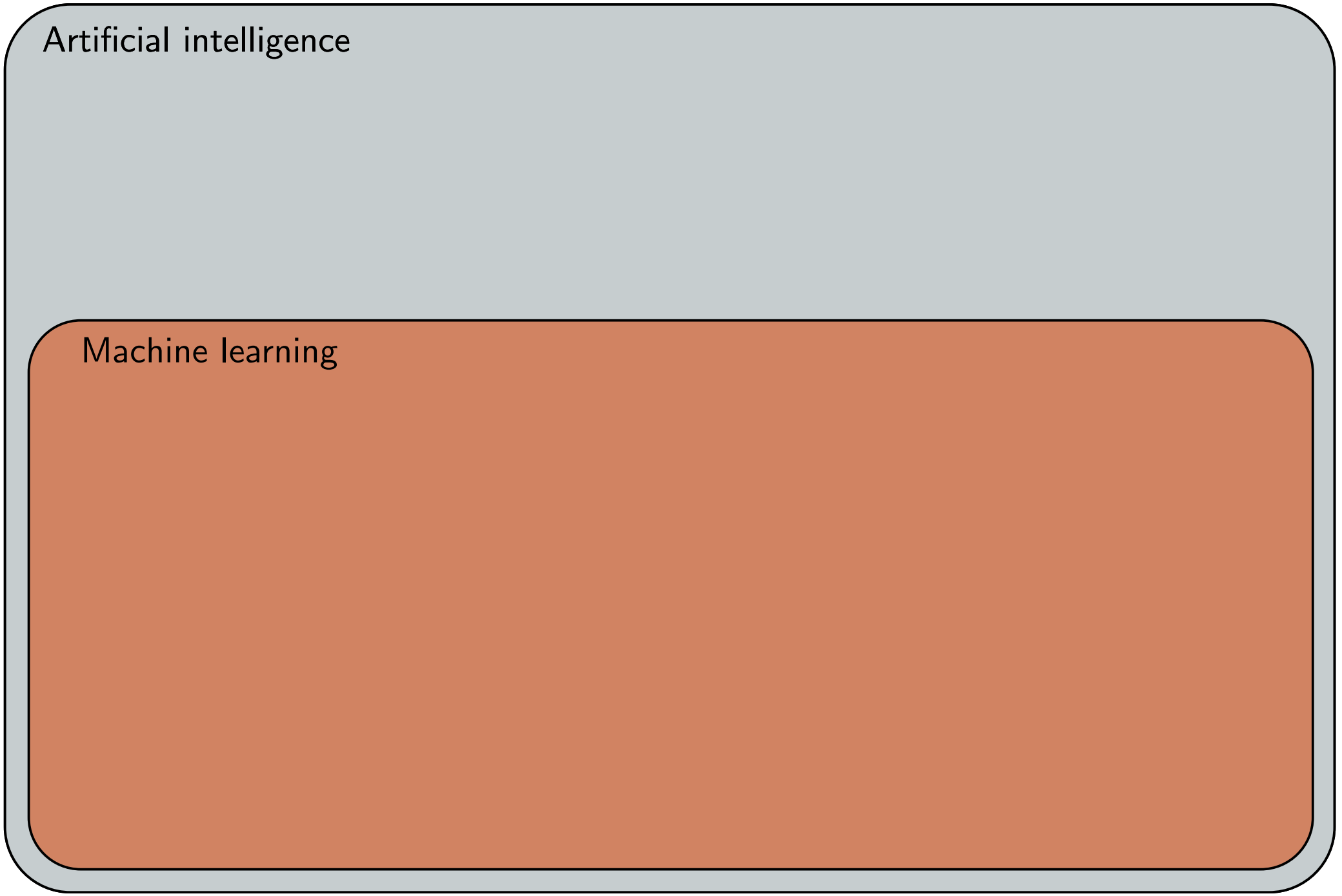
Today

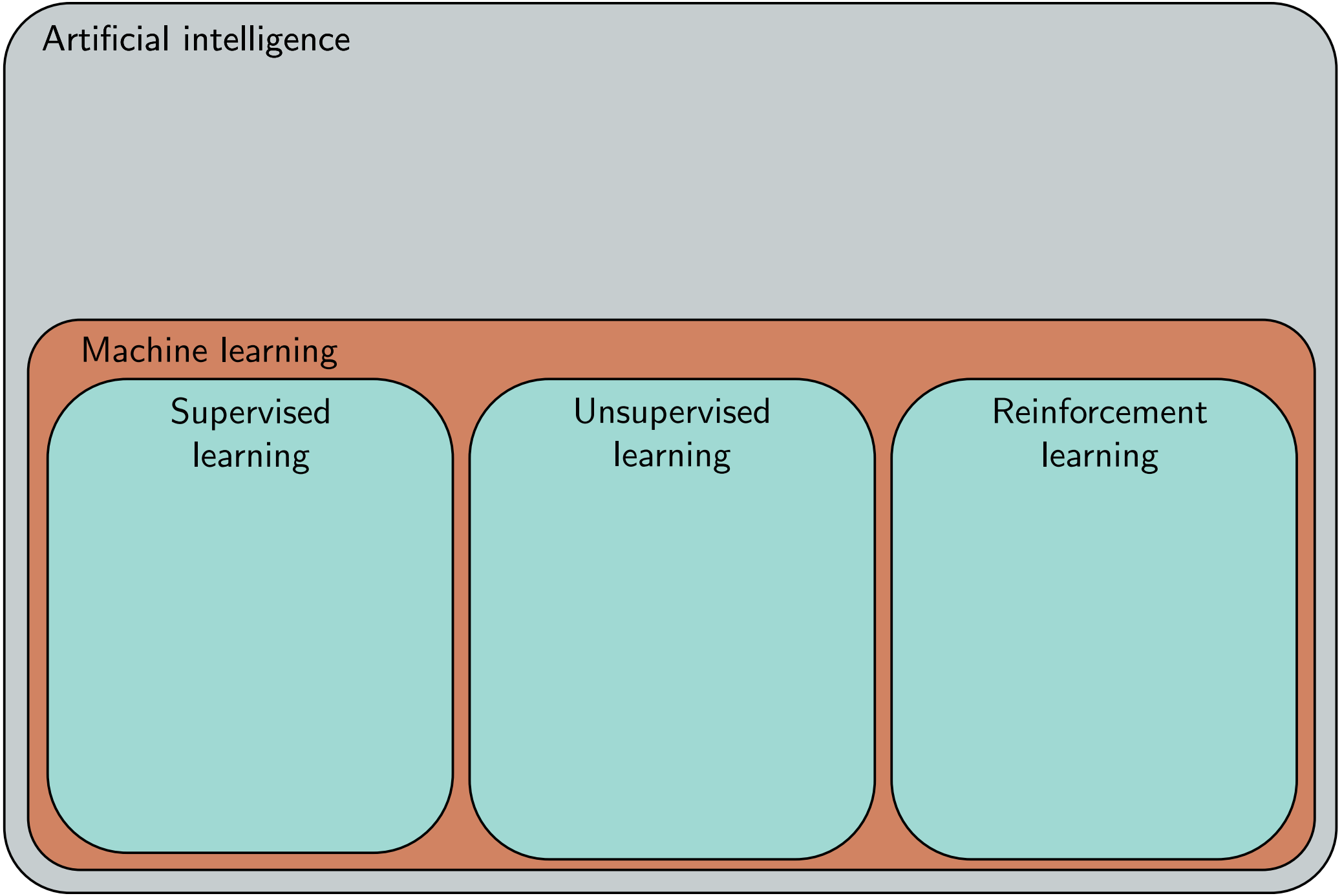
- Introduction to and Applications of Deep Learning
- History of Neural Networks
- Course Logistics

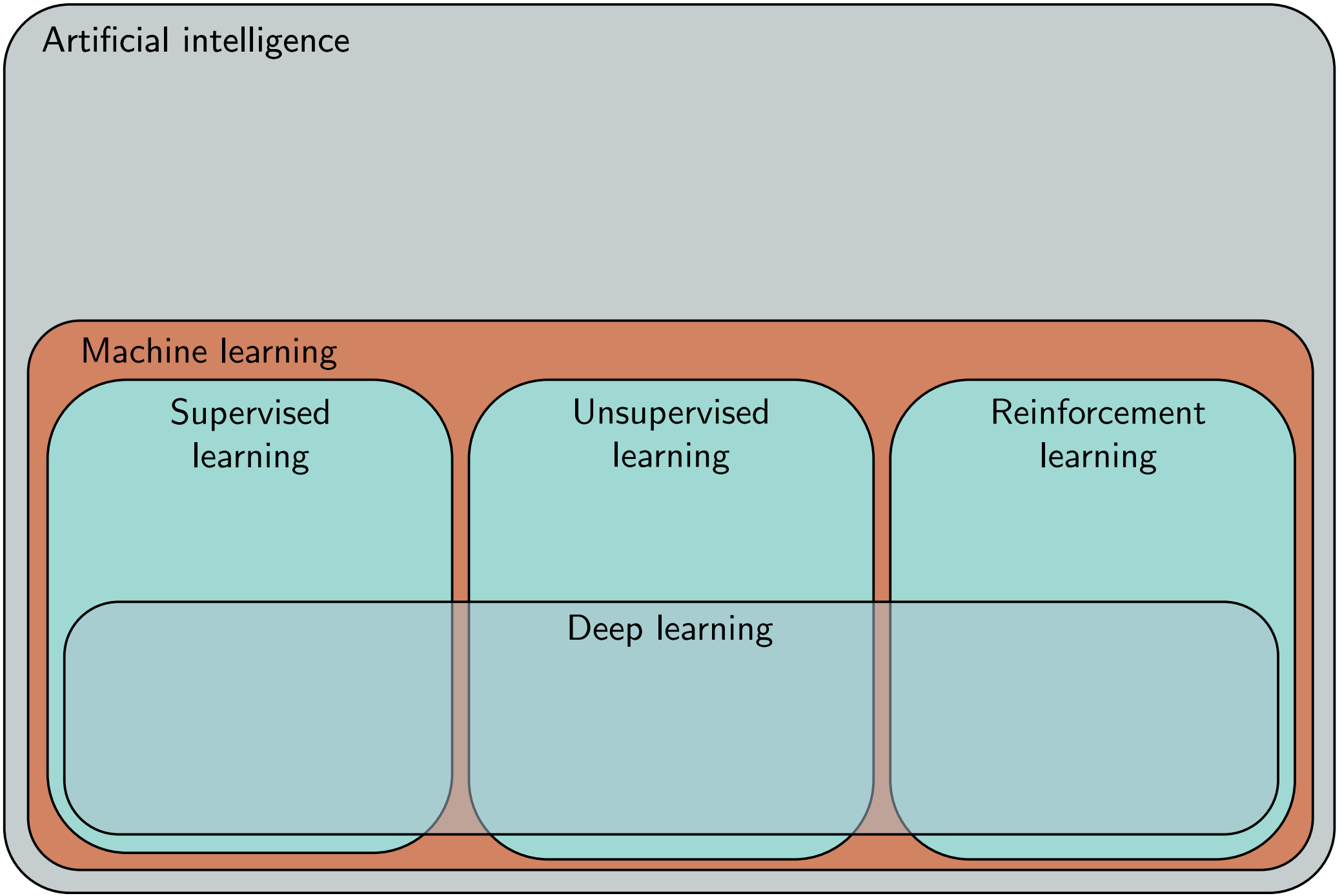
Introduction

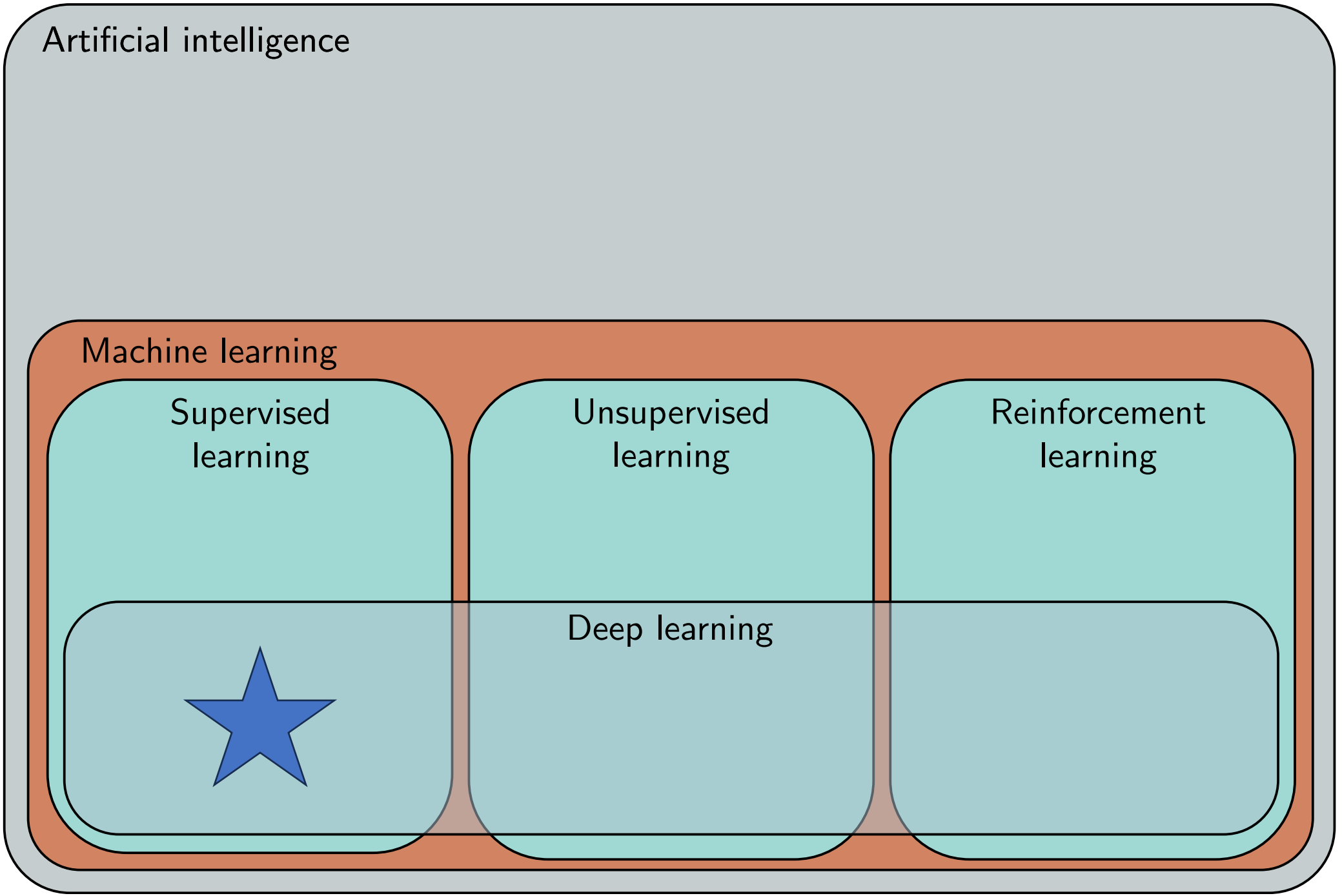
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Artificial intelligence


Artificial intelligence
Machine learning

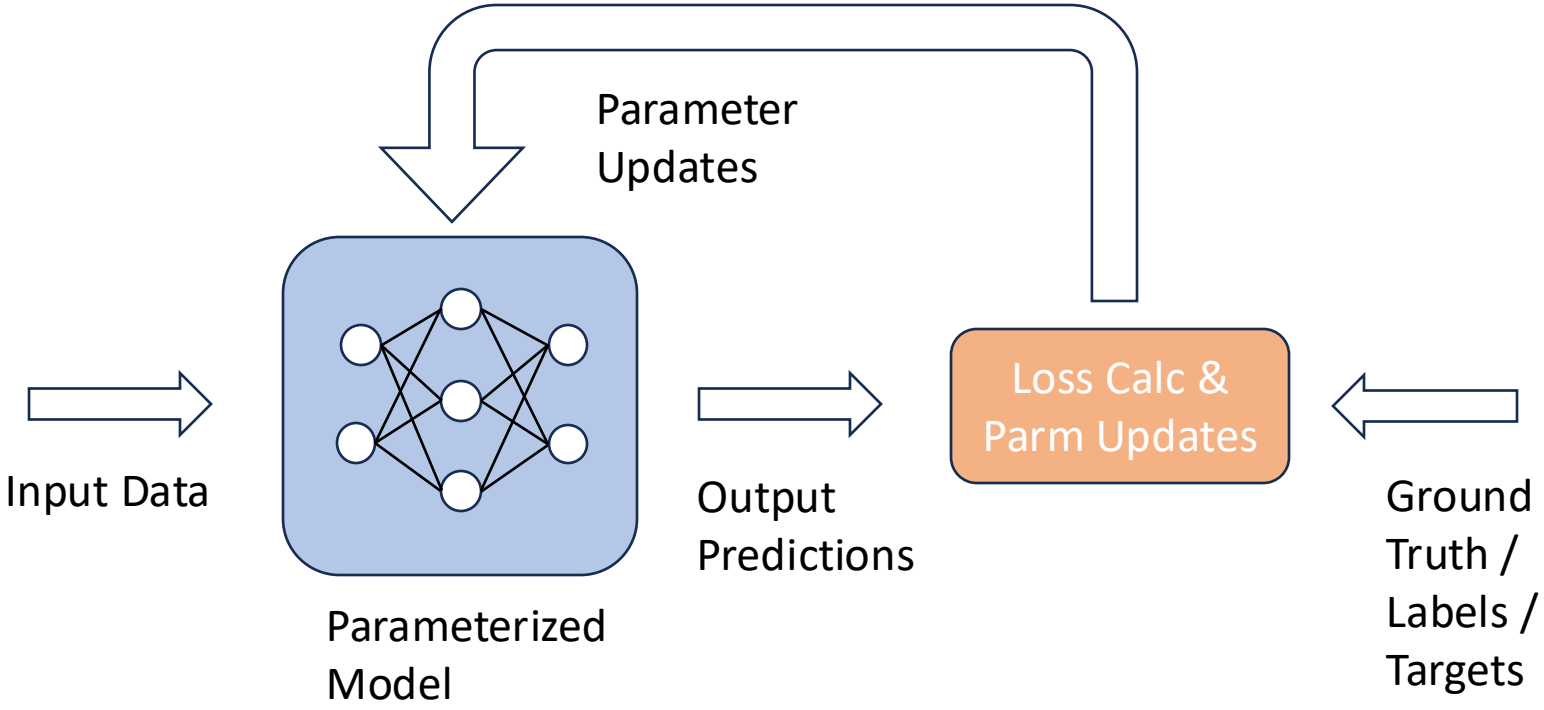

Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning


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

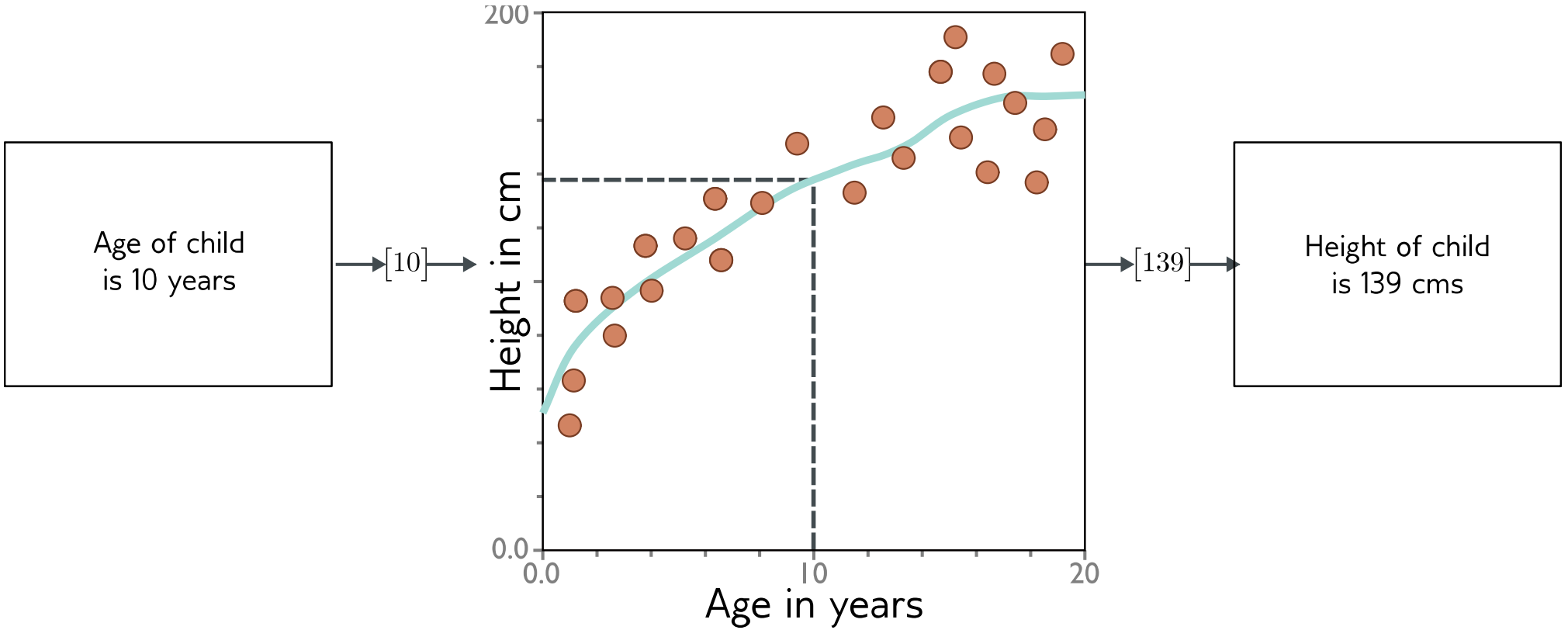

Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Supervised learning

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

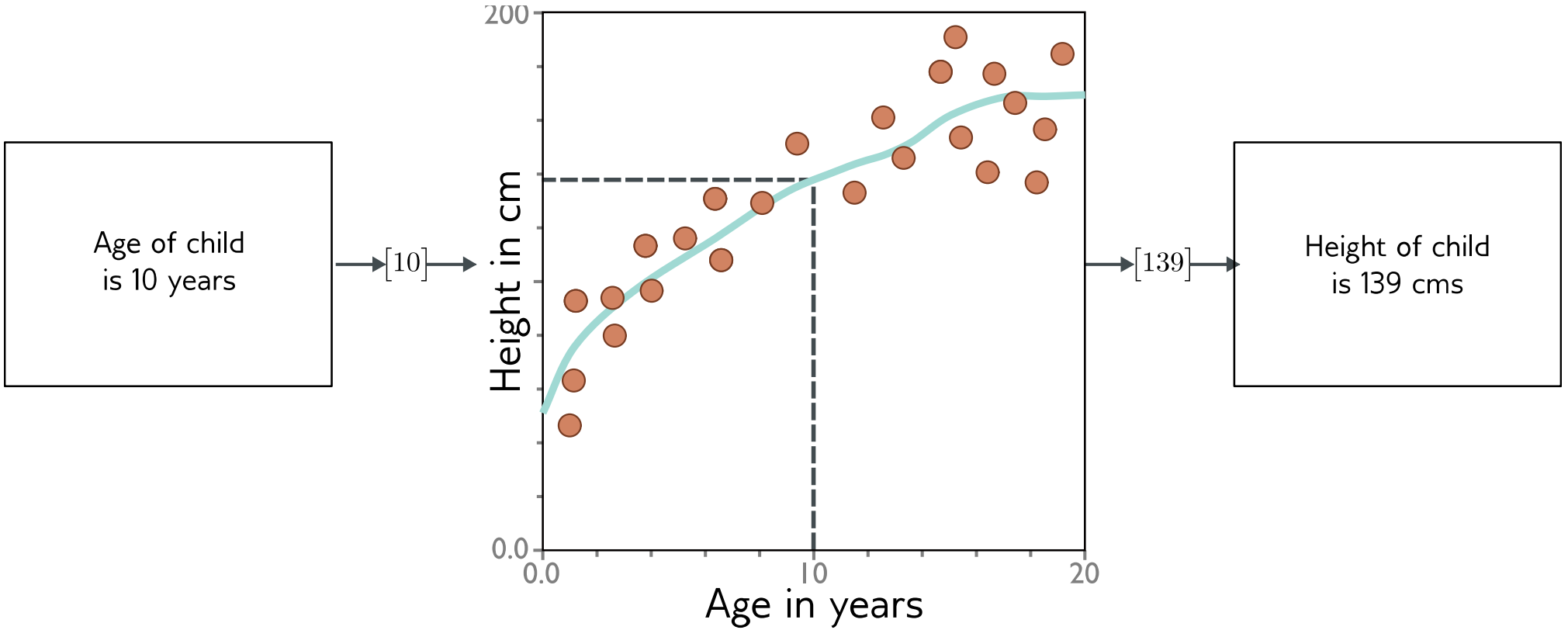


What is a supervised learning model?



- 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”

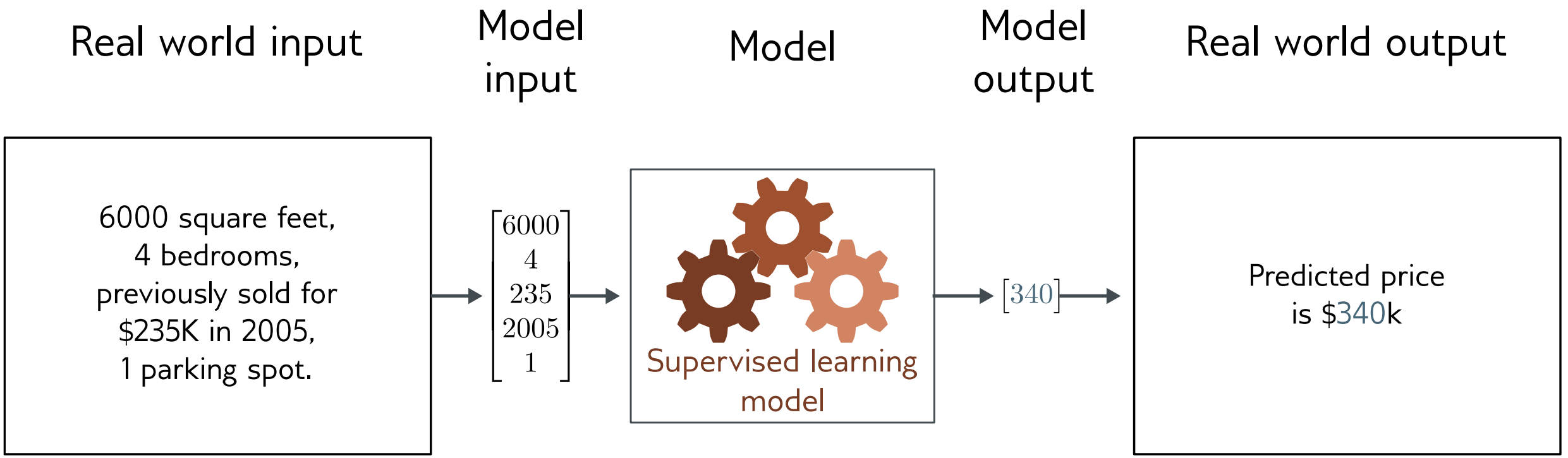
Prediction Types

- 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

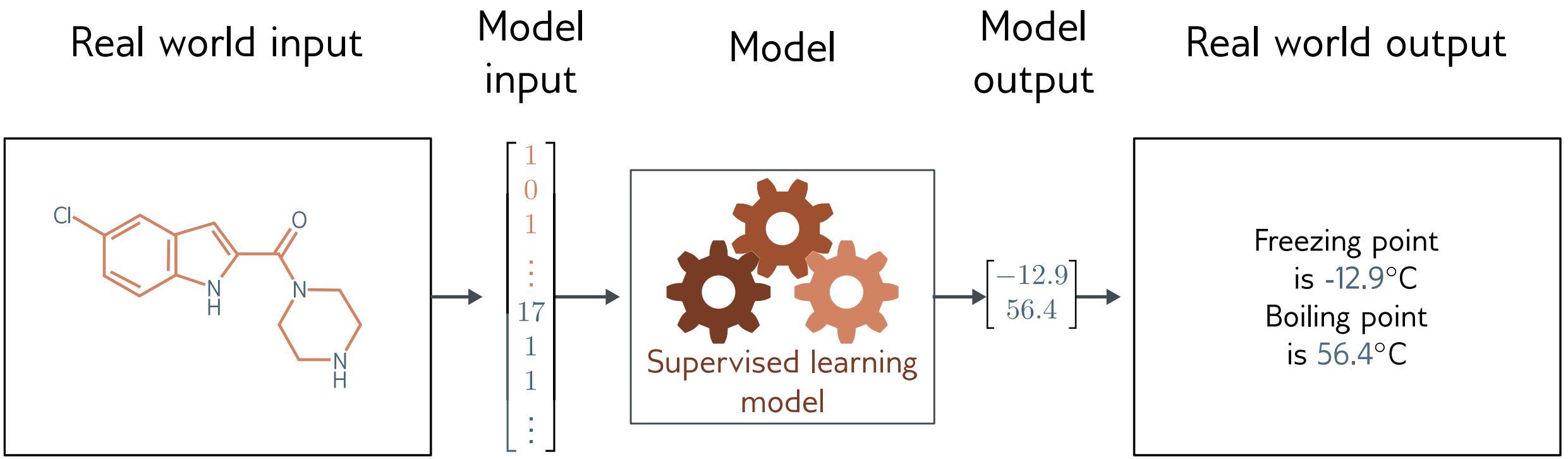
  Can be univariate (one output) or multivariate (more than one output)

Regression



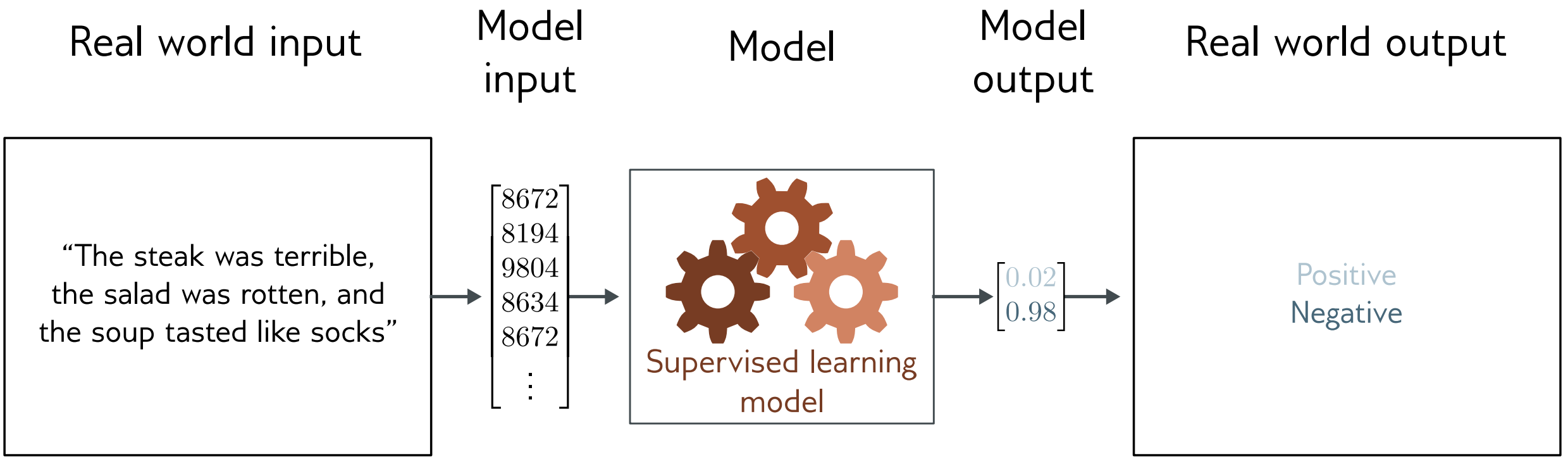
- Univariate regression problem (one output, real value)
- Fully connected network

Graph regression



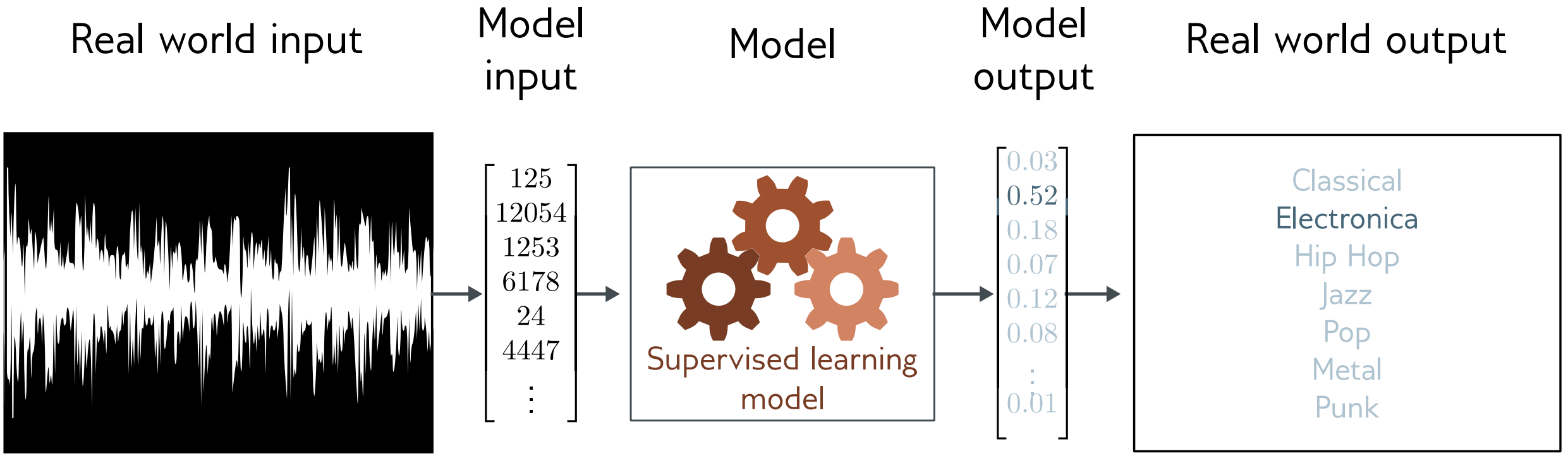
- Multivariate regression problem (>1 output, real value)
- Graph neural network

Text classification



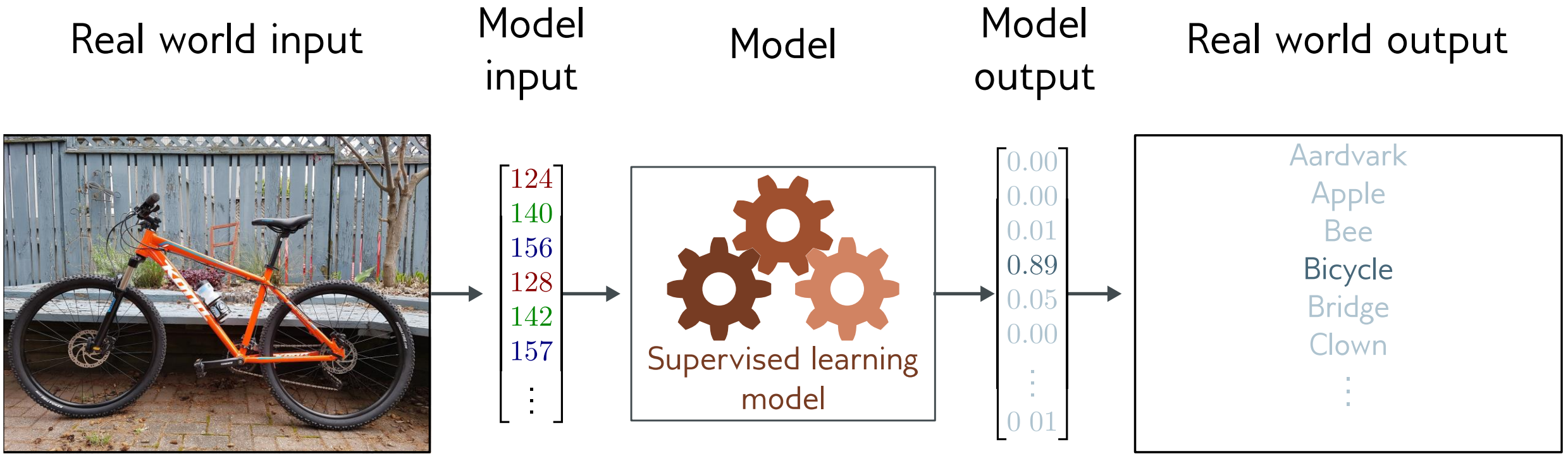
- Binary classification problem (two discrete classes)
- Transformer network

Music genre classification



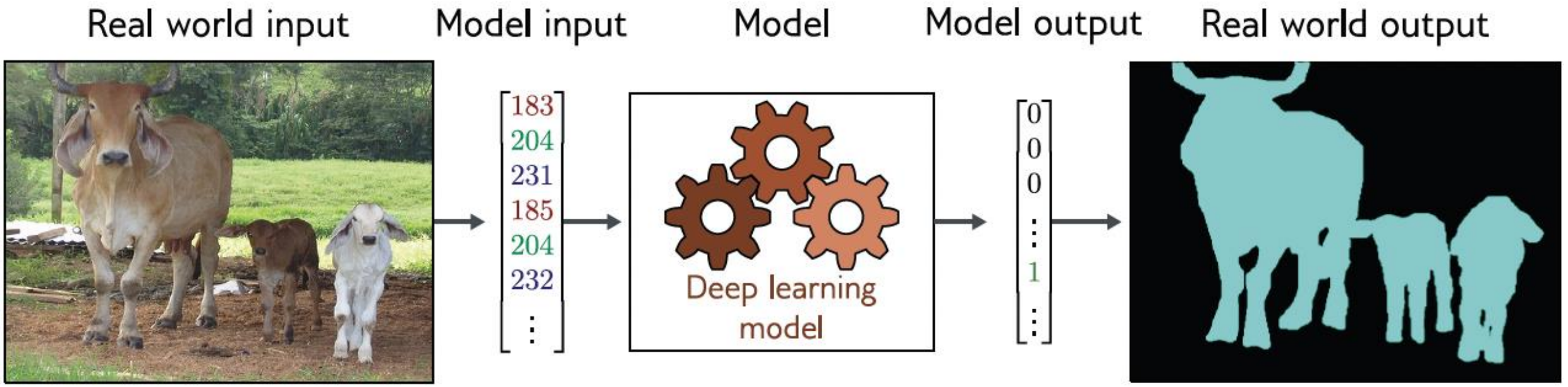
- Multiclass classification problem (discrete classes, >2 possible values)
- Recurrent neural network (RNN)

Image classification



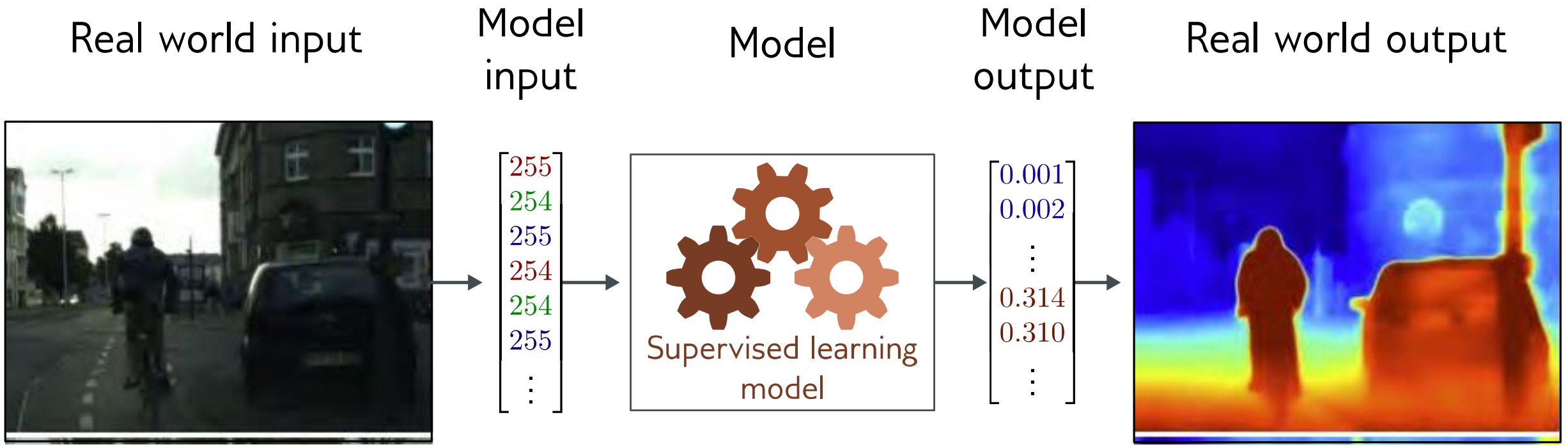
- Multiclass classification problem (discrete classes, >2 possible classes)
- Convolutional network

Image segmentation



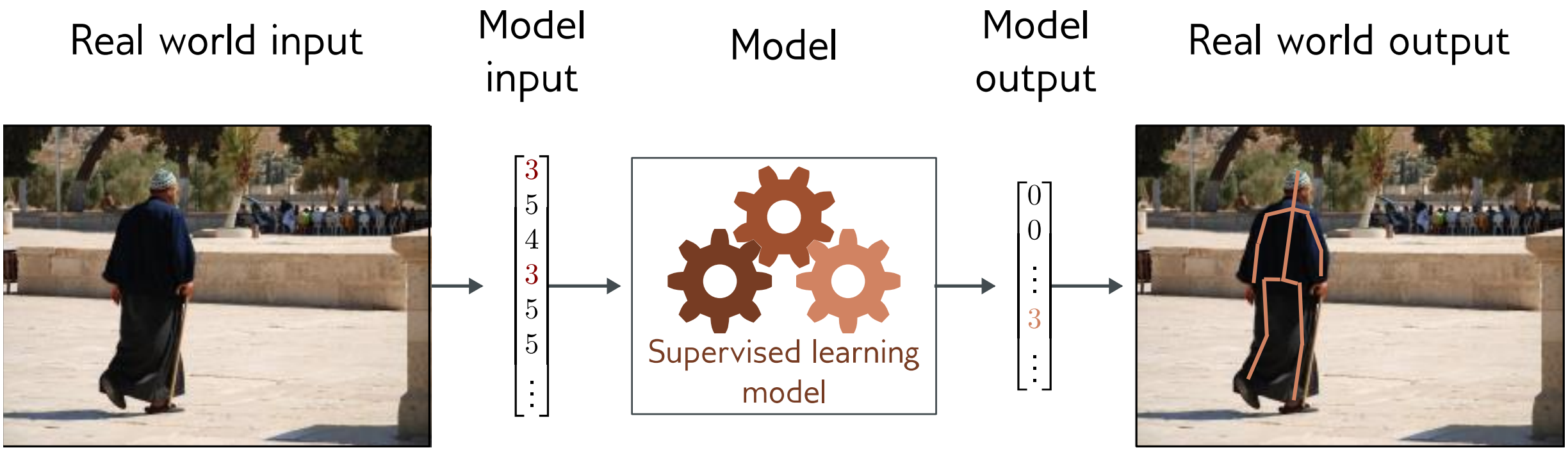
- Multivariate binary classification problem (many outputs, two discrete classes)
- Convolutional encoder-decoder network

Depth estimation



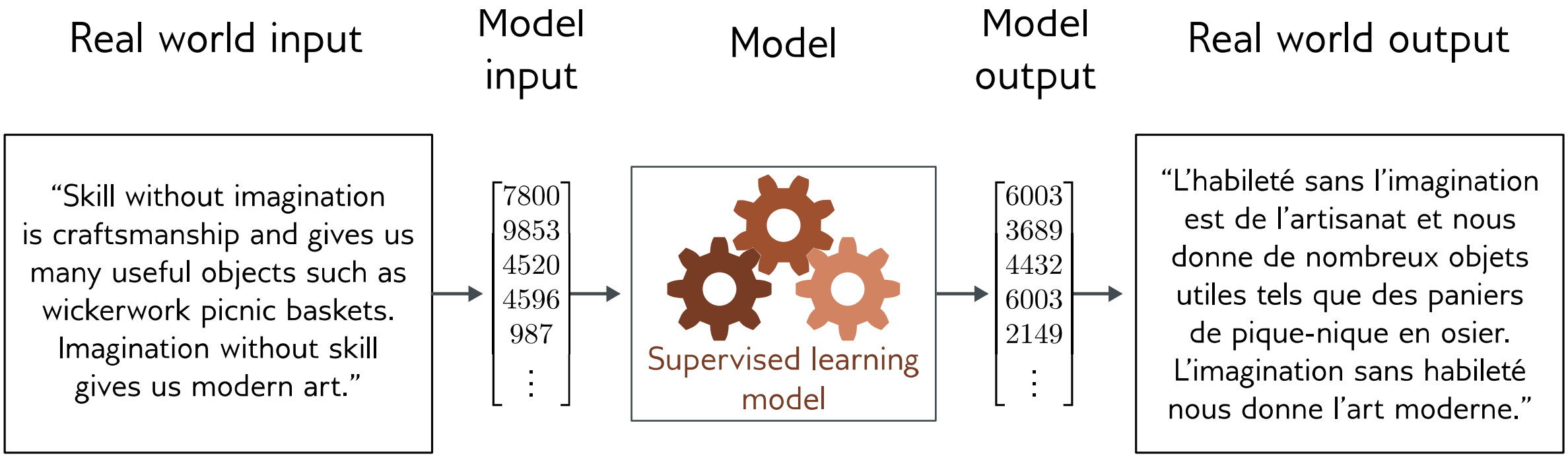
- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Pose estimation



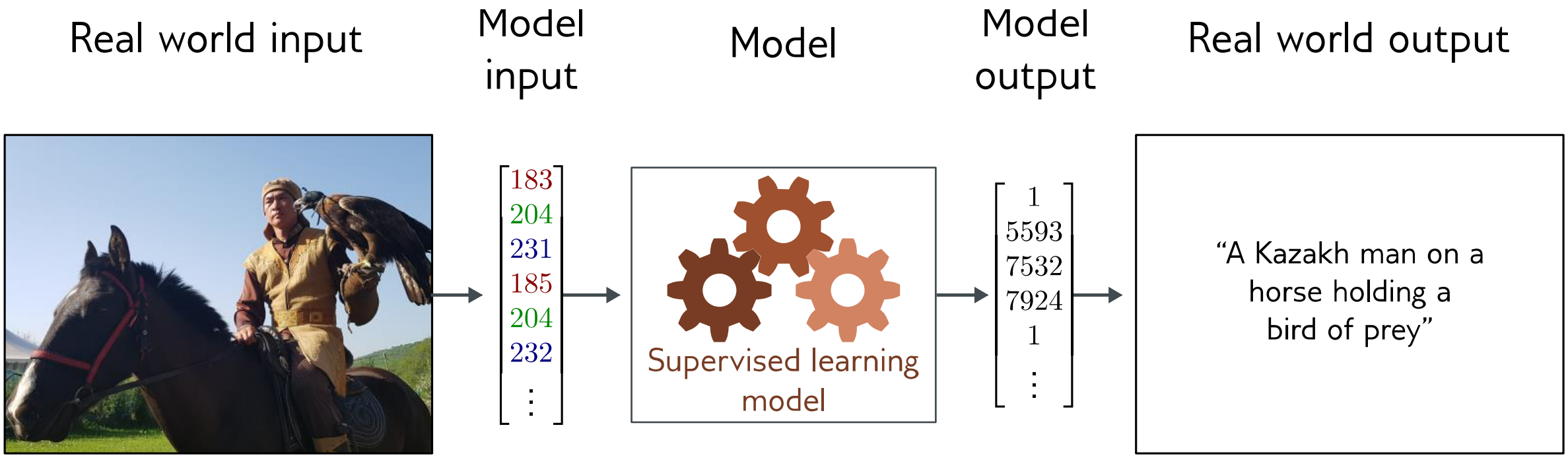
- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Translation



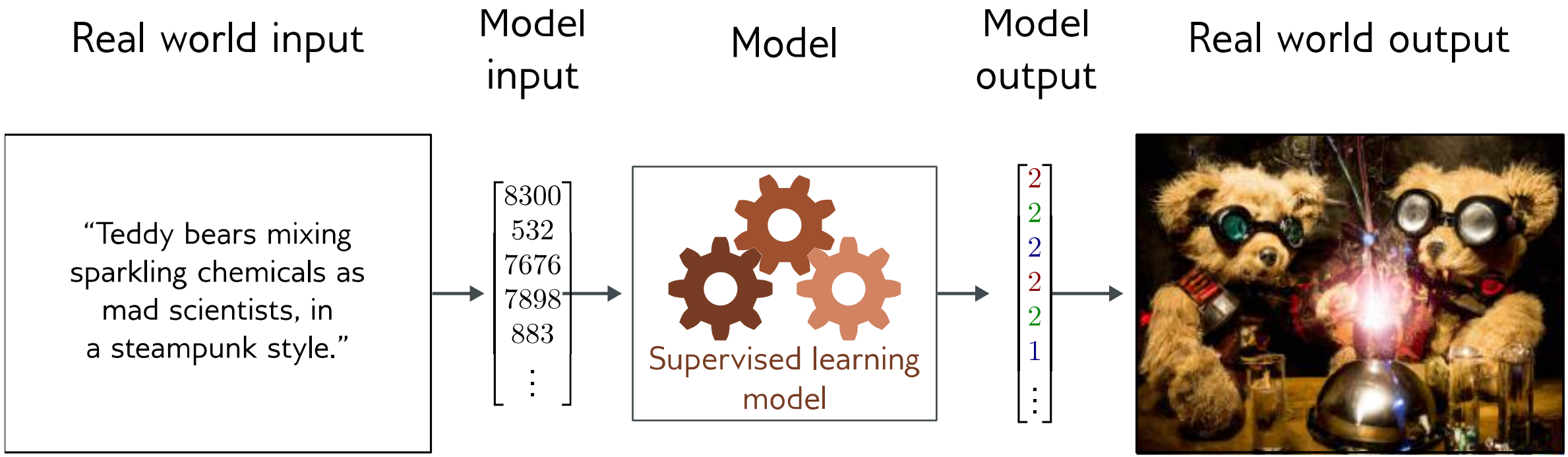
- Encoder-Decoder Transformer Networks

Image captioning



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

Image generation from text



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

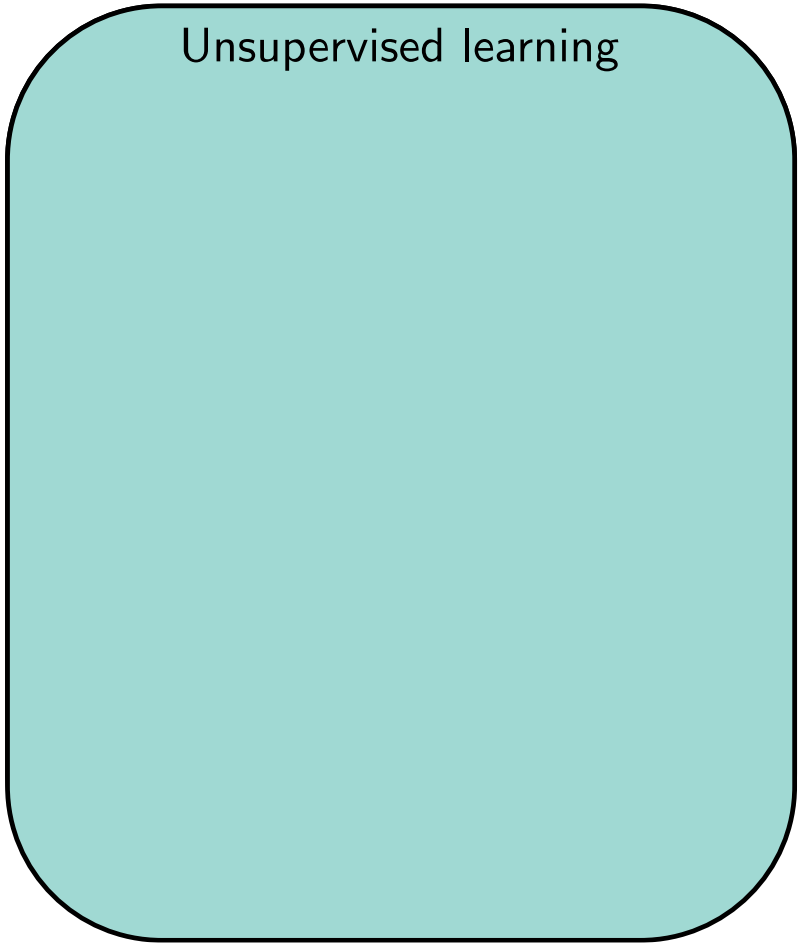
Natural images also have "rules"

Idea

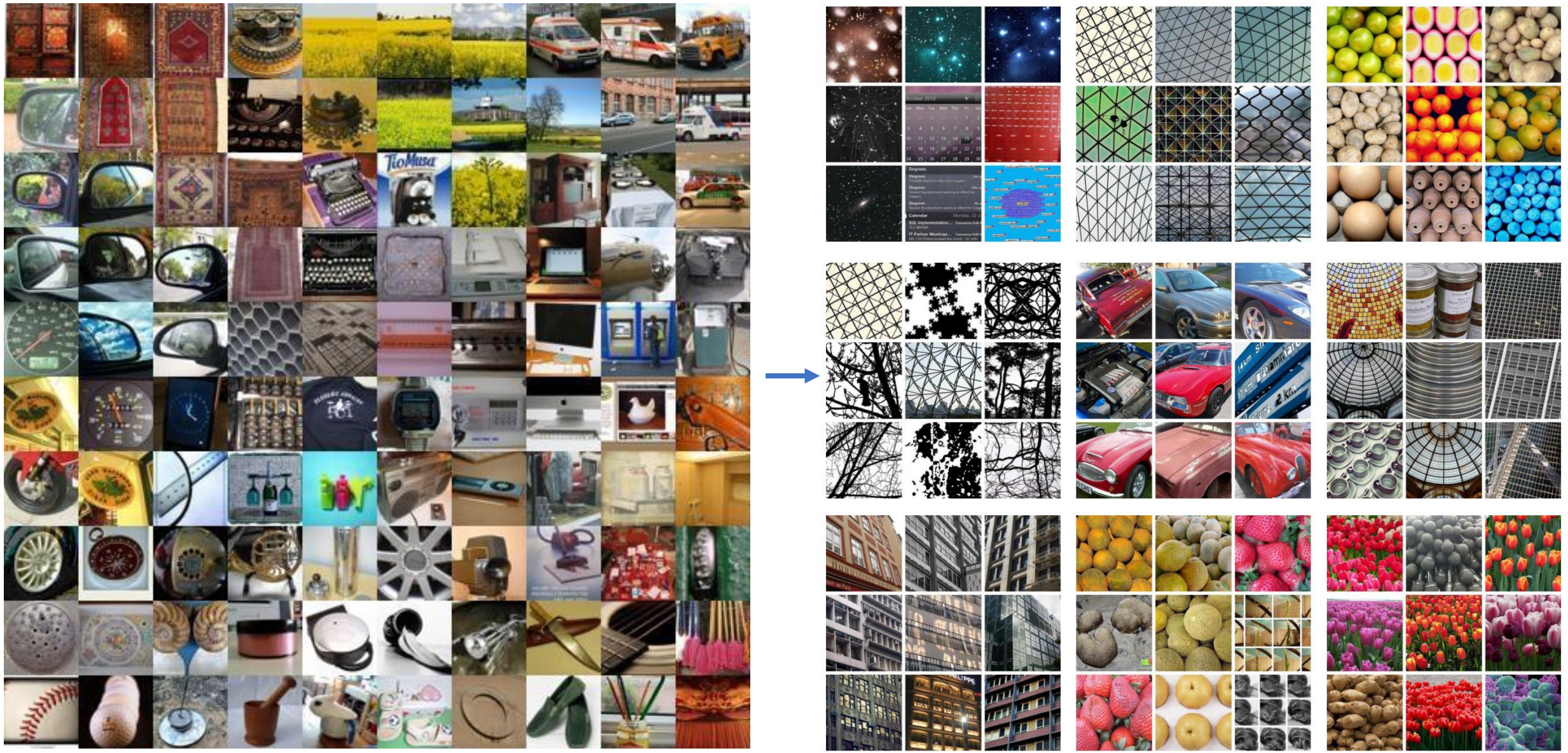
- Learn the “grammar” of the data from unlabeled examples
- Can use a gargantuan amount of data to do this (as unlabeled)
- Make the supervised learning task easier by having a lot of knowledge of possible outputs

Unsupervised Learning

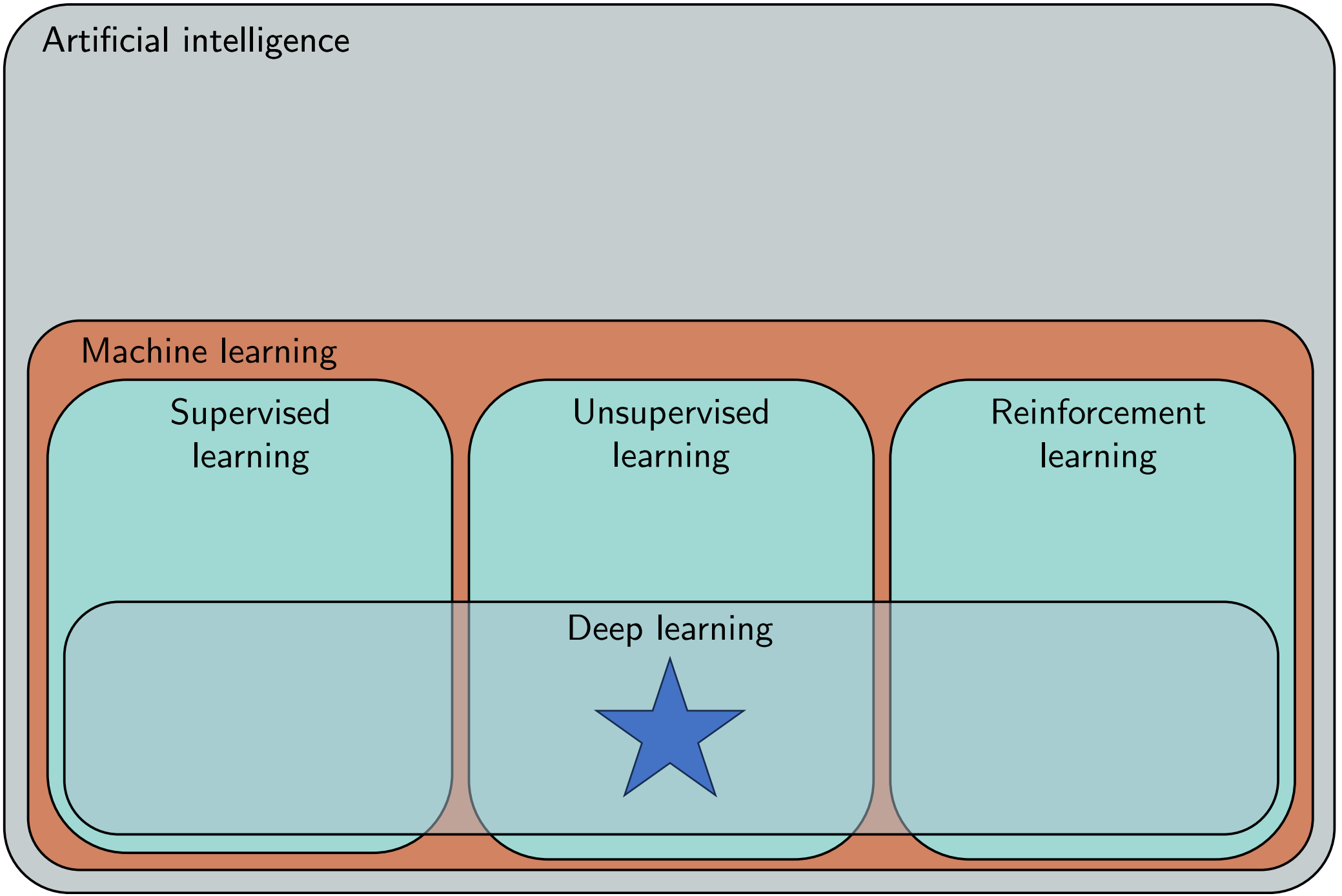
- Learning about a dataset without labels
  - Clustering
  - Finding outliers
  - Generating new examples
  - Filling in missing data



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

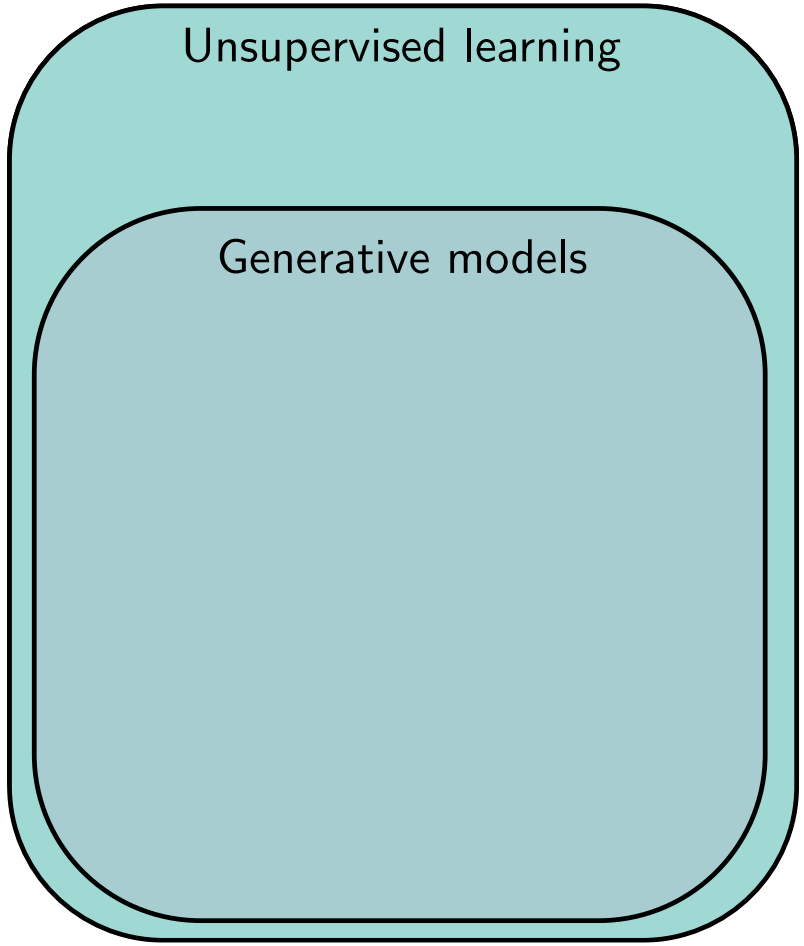


DeepCluster: Deep Clustering for Unsupervised Learning of Visual Features (Caron et al., 2018)


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

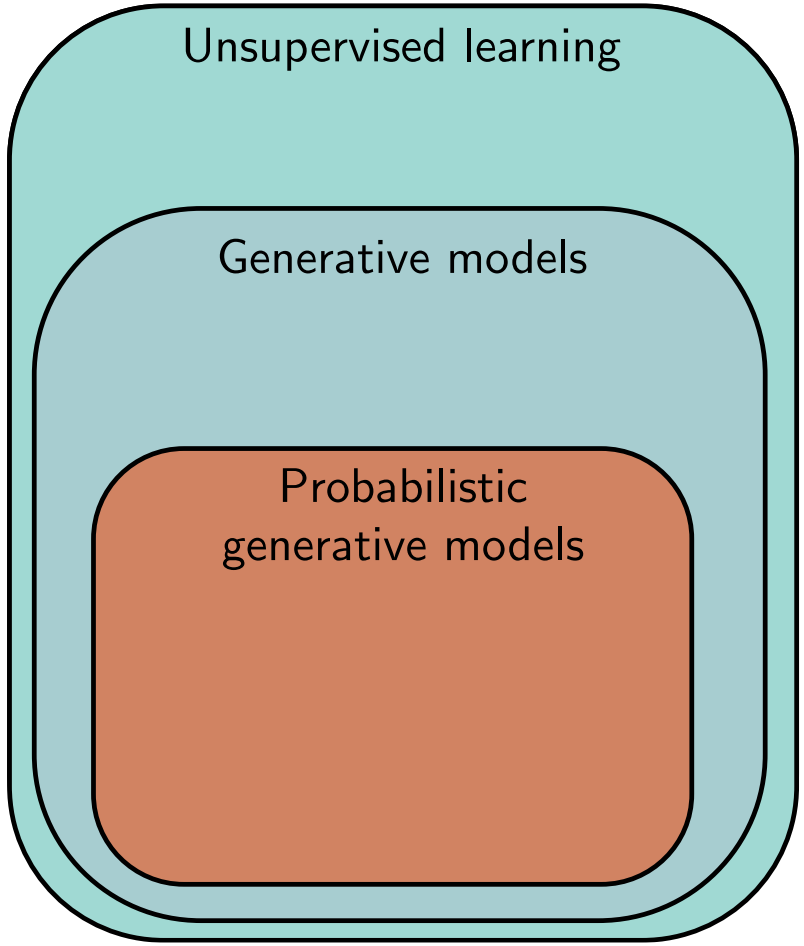
Unsupervised Learning

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

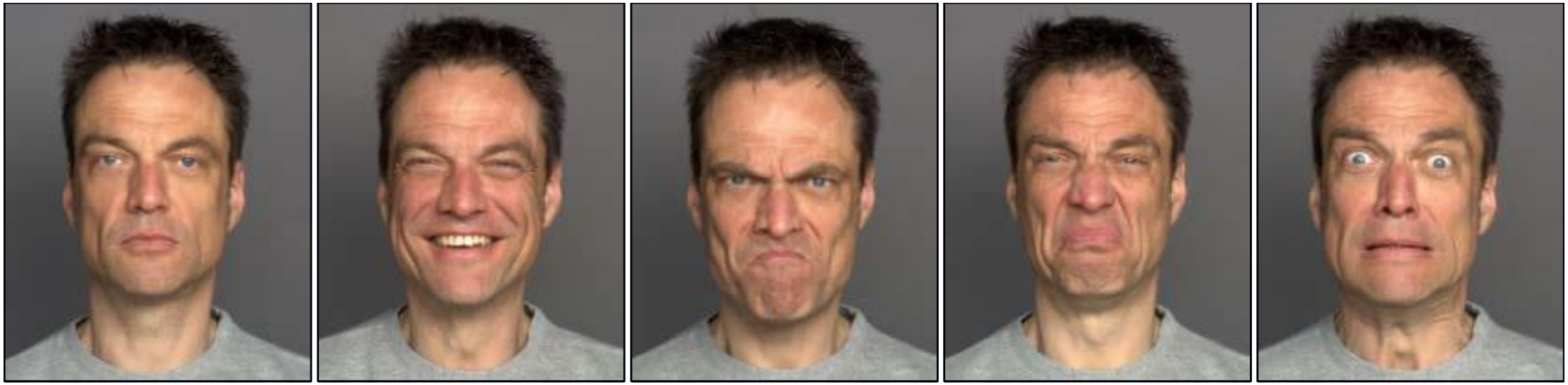


Unsupervised Learning

- 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

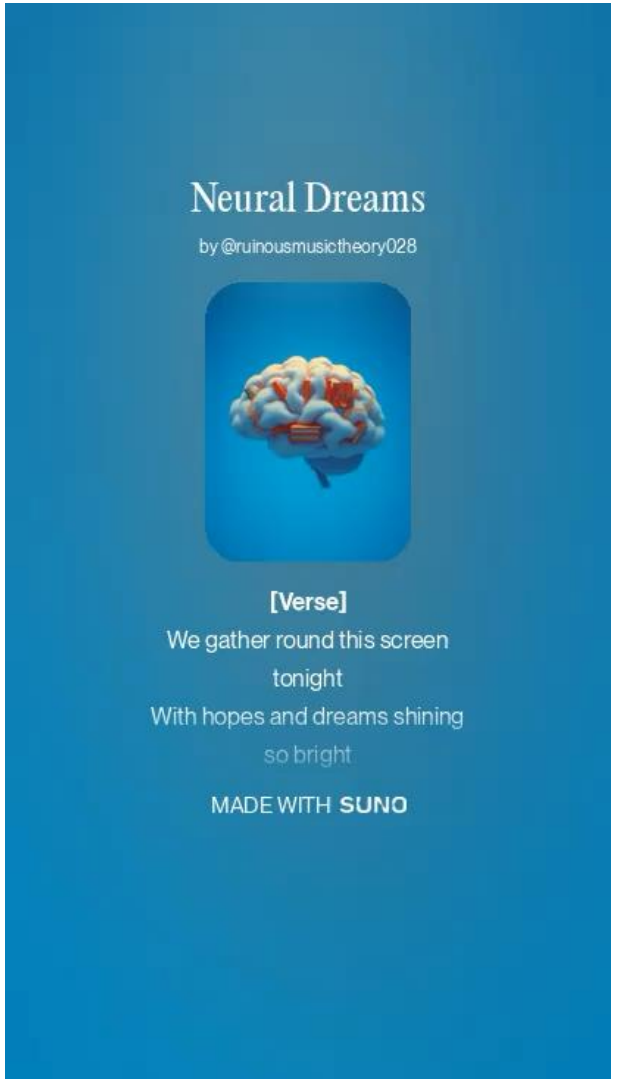
Conditional synthesis

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

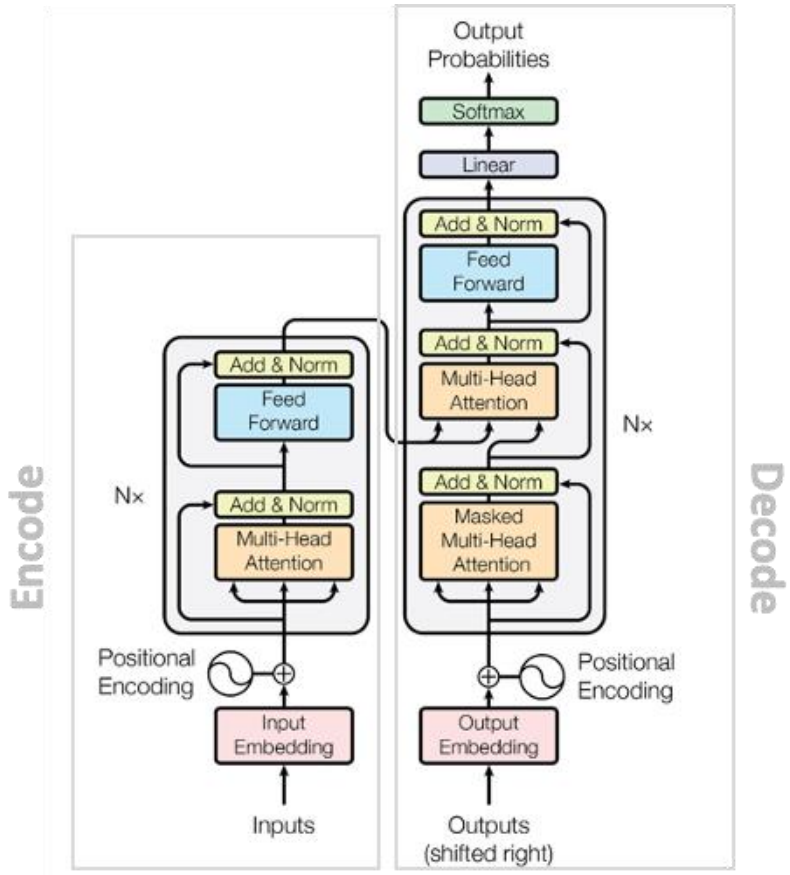
A teenage superhero fighting crime in an urban setting shown in the style of claymation.

https://sora.com

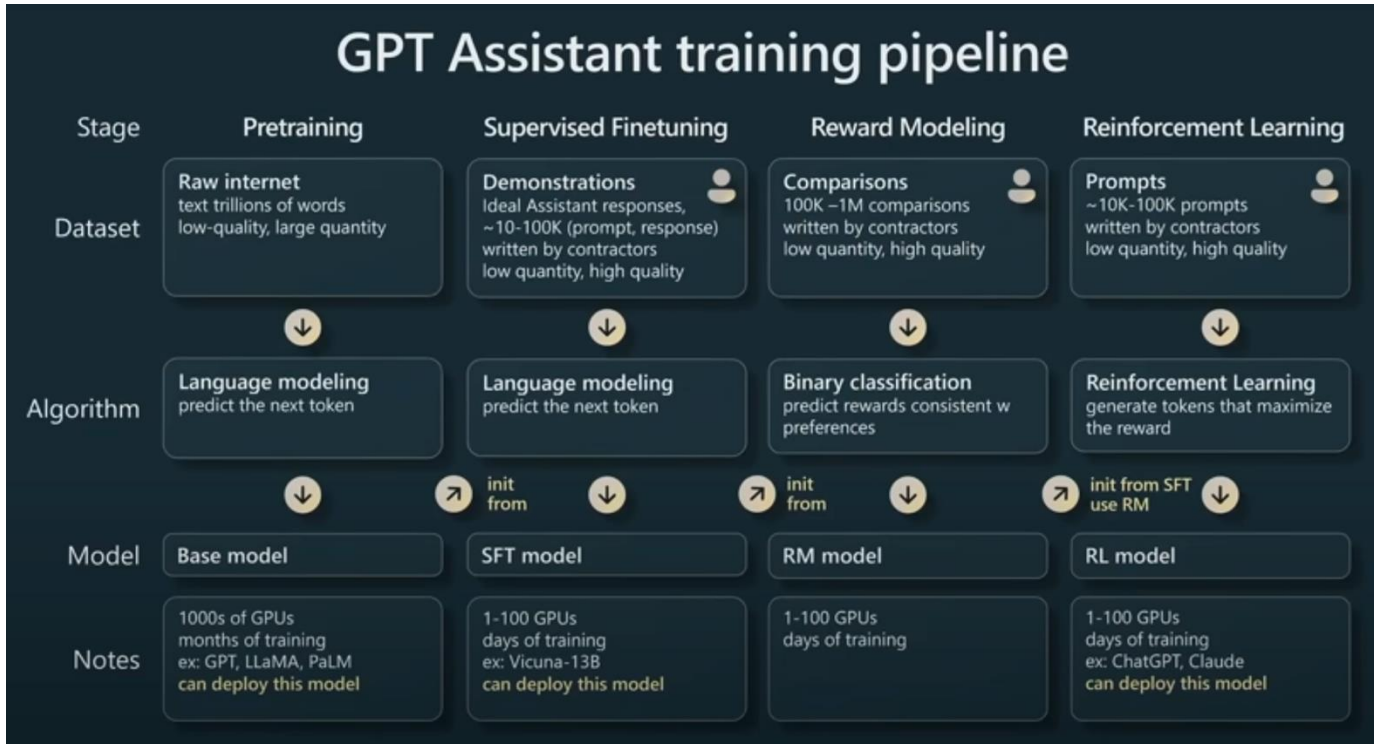


Write a short pop song about students wanting to learn about neural networks and do great things with them.

Transformers, GPTs and Assistants

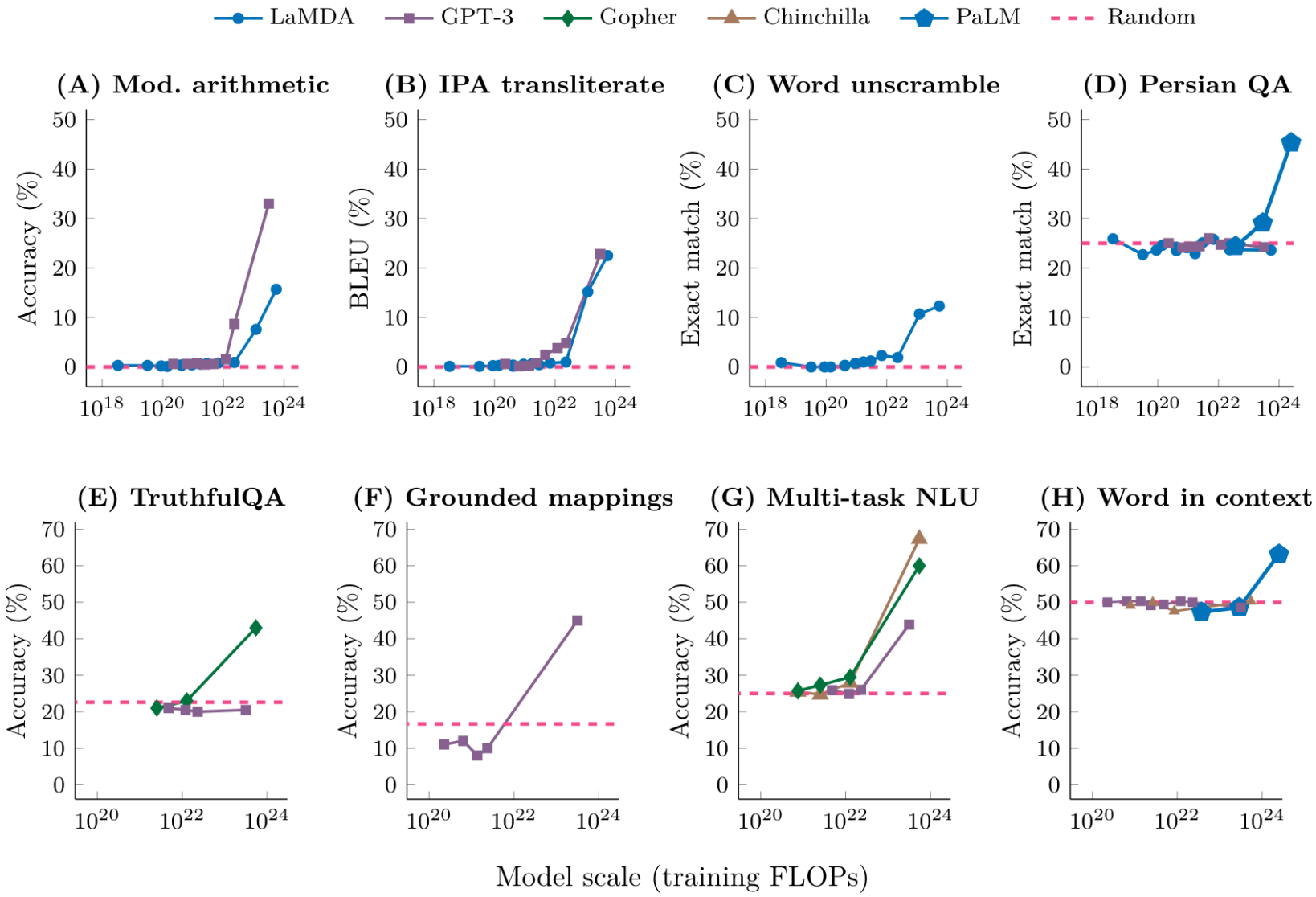


A. Vaswani *et al.*, "Attention is All you Need," presented at the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017, p. 11. [Online]. Available: https://arxiv.org/abs/1706.03762



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.

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

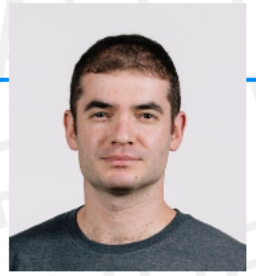
- 1979 – Fukushima: Neocognitron, cascade of neural structures that can classify shapes, invariant to shift, learned from data
- 1982 – Hopfield Networks, recurrent artificial neural networks
- 1983 – Hinton & Sejnowski: Boltzmann Machines
- 1985 – Rumelhart, Hinton, Williams: Practical backpropagation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

A Brief History of Transformers

2000

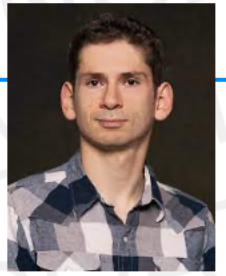
Yoshua Bengio*

2014



Ilya Sutskever*

2014



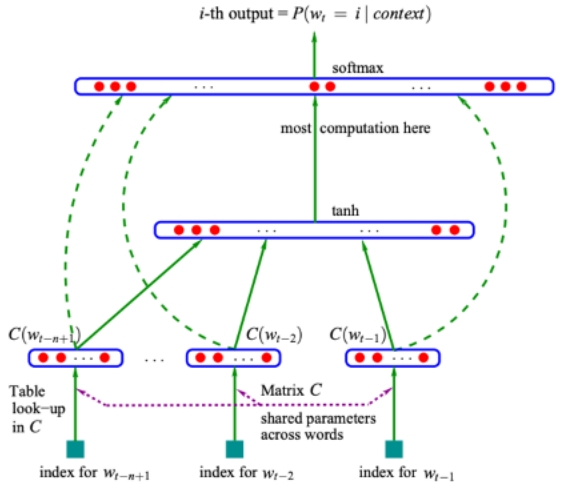
Dzmitry Bahdanau*

2017

A Team at Google

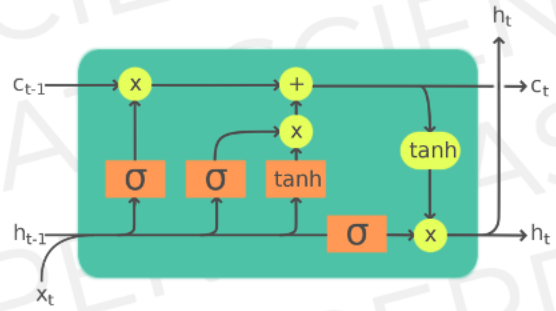
Use LSTMs

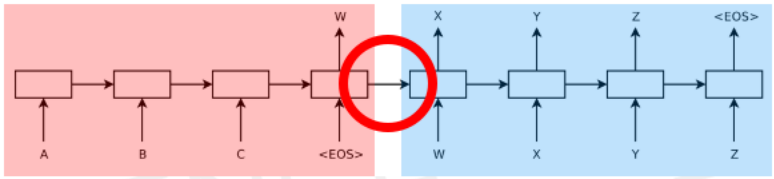
A Neural Probabilistic Language Model



Add Attention

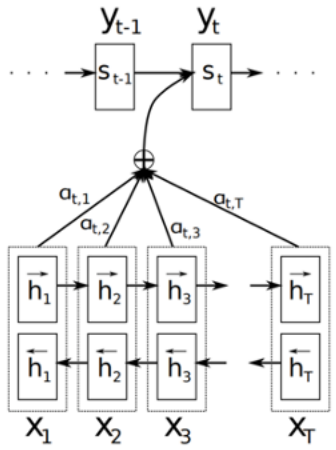
Seq-to-Seq Learning with Neural Networks



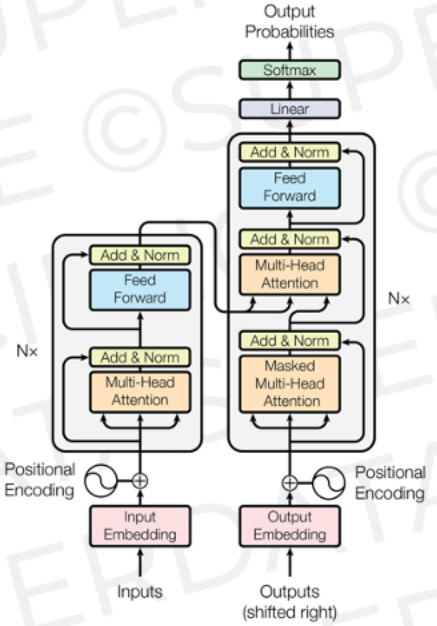


Remove LSTMs

Neural Machine Translation by Jointly Learning to Align and Translate



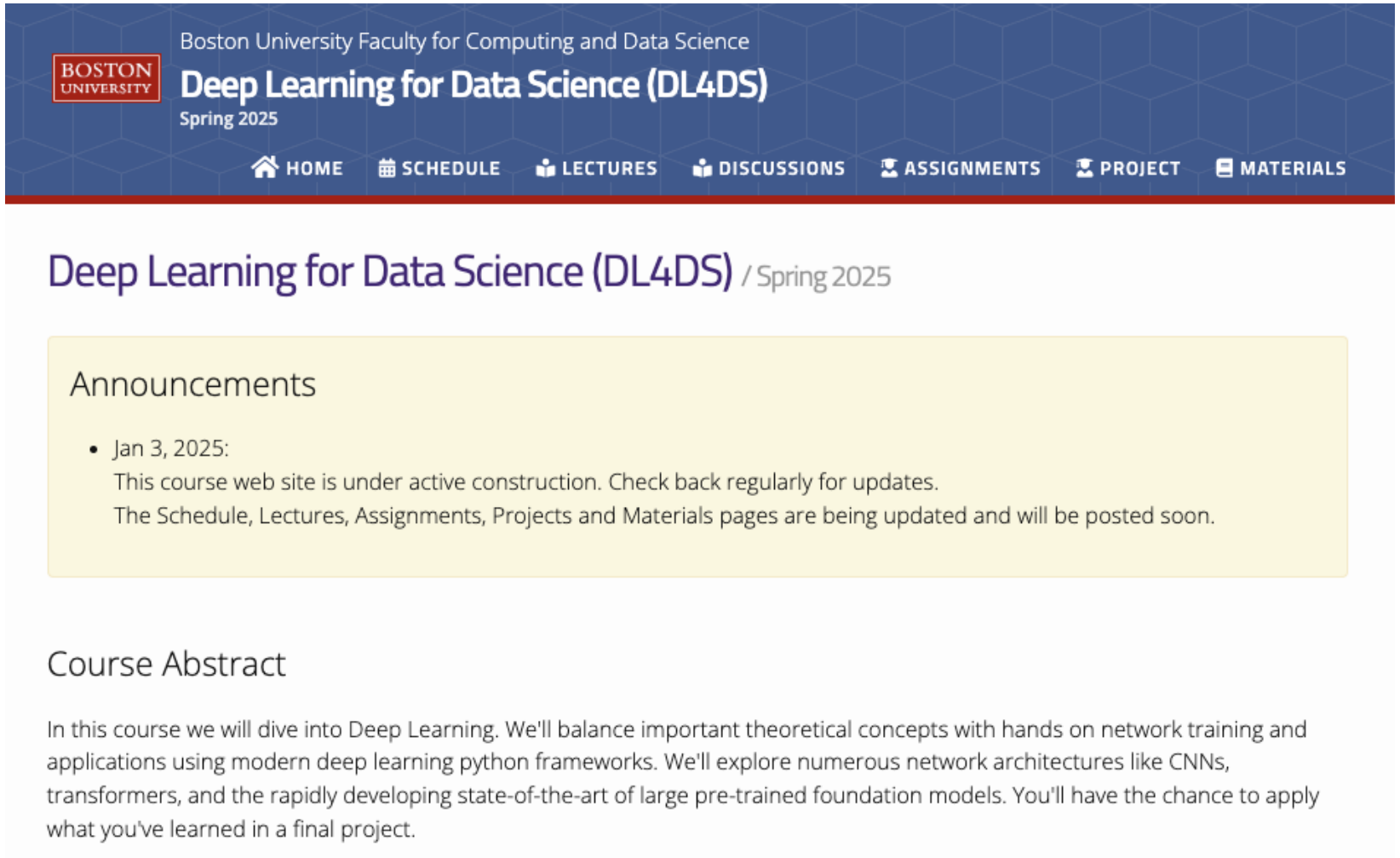
Attention is all you need



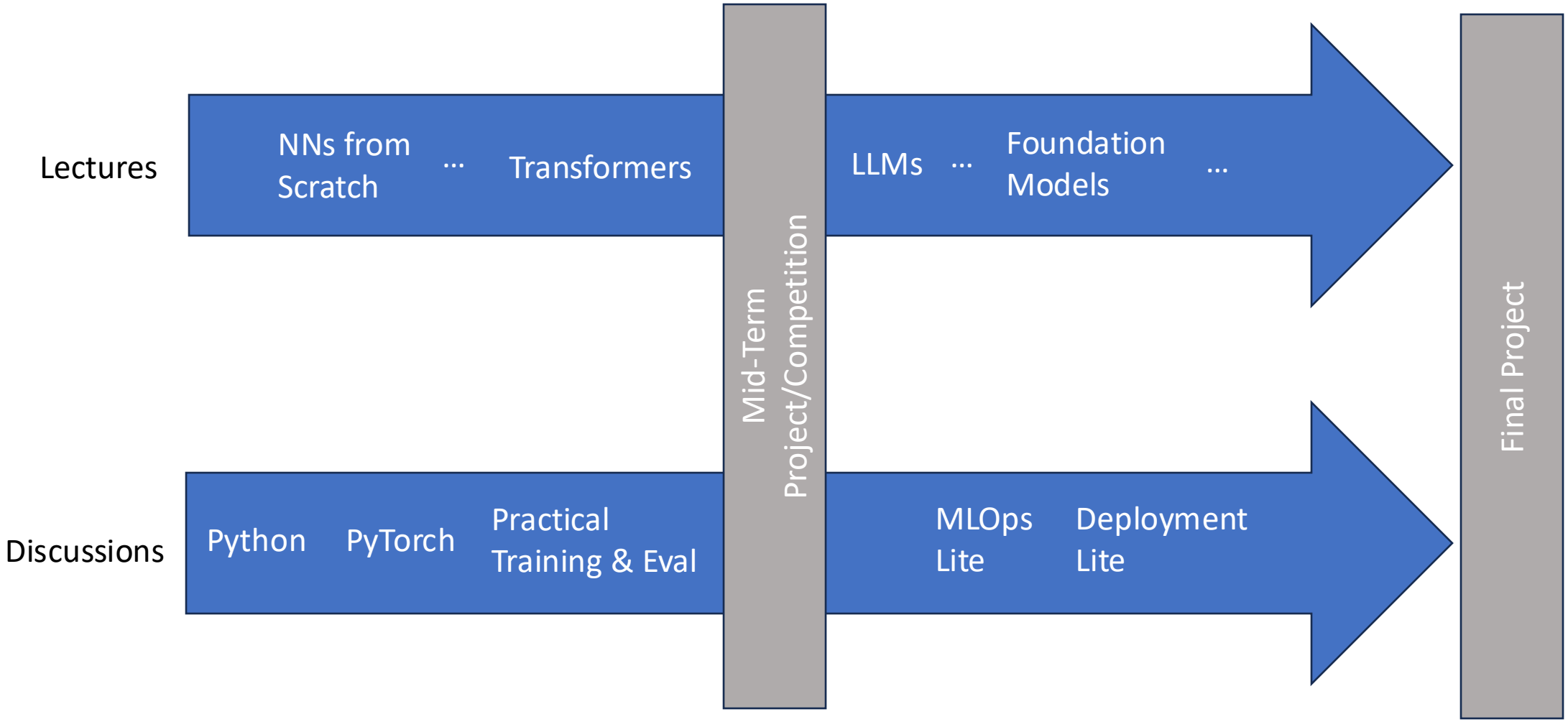
And others; Chronological analysis inspired by Andrej Karpathy's lecture, youtube.com/watch?v=XfpMkf4rD6E

Course Logistics

Course Website: https://dl4ds.github.io/sp2025/



Balancing Theory and Practice – Two Tracks



Course Outline -- Lectures

Subject to some tweaks

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. Recurrent Neural Networks
14. Transformers Part 1
15. Transformers Part 2

Python & PyTorch Practice

Spring Break

Second Half

16. Visual & Multimodal Transformers
17. Improving LLM Perf
18. Parameter Efficient Fine Tuning
19. Language Embeddings and Models
20. Unsupervised Learning and GANs
21. Variational Autoencoders
22. Diffusion Models
23. Graph Neural Networks
24. TBD
25. TBD
26. **Final Project Presentations**
27. **Final Project Presentations**

Python & PyTorch Applications

Discussion Outline*

Week	Discussion Content	Prev. Lectures
Week 1	SCC + Environment Setup	1.Intro to DL, Logistics 2.Supervised Learning
Week 2	Pytorch, Tensors, and Tensor Operations	3.Shallow Networks 4.Deep networks
Week 3	How to read, load, and process data Intro to Model Building in Pytorch.	5.Loss Functions 6.Fitting Models
Week 4	Autograd, and Computational Graphs in Pytorch. +	7.Gradients and backprop 8.Initialization
Week 5	Intro to Model training, Internals of Model training (via code) •How models run on GPUs + How to construct Eval Metrics for Different Applications	9.Measuring Performance
Week 6	CNNs + Logging, Checkpointing, Tracking Experiments, Hyperparam tuning	10.Regularization 11.CNNs
Week 7	OH for setup of Midterm Competition	12.ResNets 13.RNNs
Week 8	Transformers + Code Examples	14.Transformers I 15.Transformers II
Week 9	VLMs + LLaVA	16.Vision and Multimodal Transformers 17.Improving LLM Perf
Week 10	LoRA, PEFT, ReFT - Latest Research on these	18.Parameter Efficient Fine-Tuning 19.Unsupervised Learning GANs
Week 11	VAE	20.VAE 21.Diffusion Models
Week 12	Diffusion Models	22.GNNs 23.RL
Week 13	Huggingface, diffusers library, langchain, etc	24.TBD 25.TBD

*Subject to updates.

Grade Weighting – *No "High Stakes" Exams*

Item	Percentage
Final Project	45%
Mid-term Project/Competition	25%
Homeworks	25%
Attendance/Polls	5%

Course Project

- Will discuss more in next lecture

Homework

- Assignments approximately every week to help you check your understanding
- Jupyter notebooks and/or problem set
- Targeting release on Fridays and due a week later
- ~24 hour late due date with 10% penalty
- 7 days after grading to submit revision to recoup 75% or lost points
- 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?
- See course website for late submission and revision policy

Jupyter Notebooks / Otter Grader

- Jupyter notebooks to help ground theory with python
- We'll be experimenting with using Otter-Grader 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, *but Otter-Grader cells will not execute*
- First notebook is out to get you started… reach out with questions

Attendance/Piazza Polls

There will be (at least one) in-class Piazza poll in each lecture to:

- help facilitate in-class discussion
- serve as quick knowledge check

It:

- will be opened in class for a few minutes and then closed immediately after,
- doubles as attendance proxy,
- will be graded pass/fail -- basically submit to pass

You must be in class to complete the poll.

You get four (4) absences with no excuse required as well as valid excused absences, e.g. illness. Please email instructor, preferably in advance of class.

Piazza Polls *(wait for me to open)*

- Supervised Learning: https://piazza.com/class/m5v834h9pcatx/post/6
- Key characteristics of neural nets: https://piazza.com/class/m5v834h9pcatx/post/7
- Unsupervised Learning: https://piazza.com/class/m5v834h9pcatx/post/11

Mid-term Kaggle Competition

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

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

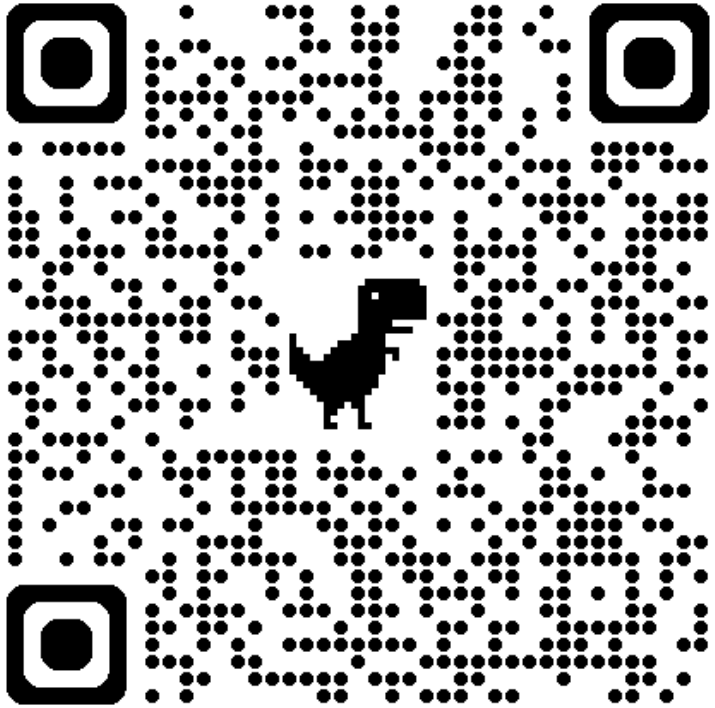
- Do the readings before the lecture – come with questions
- Stay on top of the Jupyter notebook and problem set knowledge checks
- Think about project ideas early. Talk about them with peers, advisors and instructor early and often.
- Put the time in on mid-term competition and final project… there's 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