Introduction

CSE 849 Deep Learning

Spring 2025

Zijun Cui

Partial slides are revised based on 1) CMU 11-785 Introduction to Deep Learning by Bhiksha Raj; 2) CM20315 University of Bath by Simon Prince, Georgios Exarchakis and Andrew Barnes

Logistics

- **Meetings:**
  - In person
  - Time: Tuesday and Thursday, 1pm – 2:20pm
  - Position: Urban Plan & Land Arch Bldg 8
- **Instructor:** Zijun Cui
  - Office Hour: Tuesday and Thursday, 2:20pm – 3:20pm
  - Office: 2212 EB
- **TA**: Gautam Sreekumar
  - Office Hour: TBD
  - Location: TBD

Communication

- Piazza
  - This term, we will be using Piazza for class discussion.
  - Lecture slides will be posted on Piazza after each class.
  - Sign up at: https://piazza.com/msu/spring2025/cse849

- D2L
  - Assignment release and submission will be handled through D2L
  - unless otherwise instructed.

Course Philosophy

- **No student left behind**
  - In our ideal world, each of you would earn an A
- **Please use the available resources**
  - TA
  - Me (office hours)
  - Your classmates or lab mates
- **If you’re under stress or unable to perform, please reach out**
  - To TA
  - To Me
  - We will do our best to help you

Computational Resources

- **Preferred Programming Platform**:

  Python and Pytorch

- **Computational Resources**:
  - MSU HPCC
  - Github Codespaces
  - Google Colab
  - Personal/Lab Computers

Course Objectives

- Class Description:

"This course provides a comprehensive introduction to deep neural networks. Major topics include multilayer perceptrons, convolutional neural networks, practical aspects of model creation from scratch and training, sequence modeling with recurrent neural networks and transformers, and generative probabilistic modeling. Advanced topics, including Bayesian deep learning, will also be explored. "

- By the end of the course, you will be able to:
  - Understand the math behind deep learning
  - Build your own neural network
  - Solve benchmark tasks

Pre-requisites

- A good background in linear algebra, probability, and statistics
- A good background in machine learning
- Proficiency in Python

- **Important Notes:**
  - I assume you are already familiar with these topics.
  - I will not cover them in lectures.
  - I may briefly mention some key concepts—not in a surrogate way.

Lecture Style

- **I will use many slides**
  - I will prepare all the content in slides and teach through them

- **I emphasize foundations – you will see a lot of math**
  - If you're *only* looking to learn practical techniques, this is not the class for you

- **For deriving equations**
  - I will use both symbolic derivations and visual animations

- **For hands on experience**
  - you will mostly gain it through projects

Attendance and Engagement

- **I will use in-class polls or quizzes for class participation**
  - Polls posted at random times during class
  - Or a quiz which will happen randomly during the term
  - Submit your responses on paper at the end of the class
  - We won’t score you on correctness, only on whether you submitted

- **I encourage interactions**
  - Questions are welcomed – no question is silly/wrong/embarrassing
  - I will randomly call on an ID. If your ID is called, you must answer the question
    - “I don’t know” is an acceptable answer
    - Some questions may not have clear answers and are open to discussion

- **Please participate**

Phones or Laptops

- **Keep your smart devices muted**
  - Phone rings or vibrations can be a distraction to others.
- **You can use your phones or laptops only to:**
  - View lecture slides
  - Take notes
- **This is not meant to a time for you to do your research or other tasks**

Grading Policies

- Your final performance will be graded based on the following:
  - **Attendance: 5%**
    - 1 point off per each absence
    - Total points = number of classes with polls or quizzes
  - **Homework: 15%**
    - Released and submitted through D2L
  - **Four class projects: 80%**
    - 20% per project
    - More details will be provided later

Grading Policies – Late Submission

- Late Submission:
  - Assignments are due on the specified due date.
  - Late submissions without an approved extension will receive a **10% deduction per day late**.
  - Each student has **a total of 5 free late days for the semester**, which can be used across assignments as needed.

- **Watch out:**
  - It is your responsibility to remember the due date. Excuses like “I forgot the due date” or “I remember a wrong date” are NOT acceptable.
  - It is your responsibility to leave sufficient time for submitting. Don’t wait until the last minute, as this will likely cause delays.

AI Disclosure

- You are welcome to use AI tools to assist in your studies.
- However, you are responsible for the work you submit.
- For each submission, include an AI disclosure indicating how AI has been used to help with the assignment.

Grading Rubric

Range	Grade
(86, 100]	4.0
(79, 86]	3.5
(72, 79]	3.0
(65, 72]	2.5
(55, 65]	2.0
(48, 55]	1.5
(40, 48]	1.0
(00, 40]	0.0

Any questions before we start?

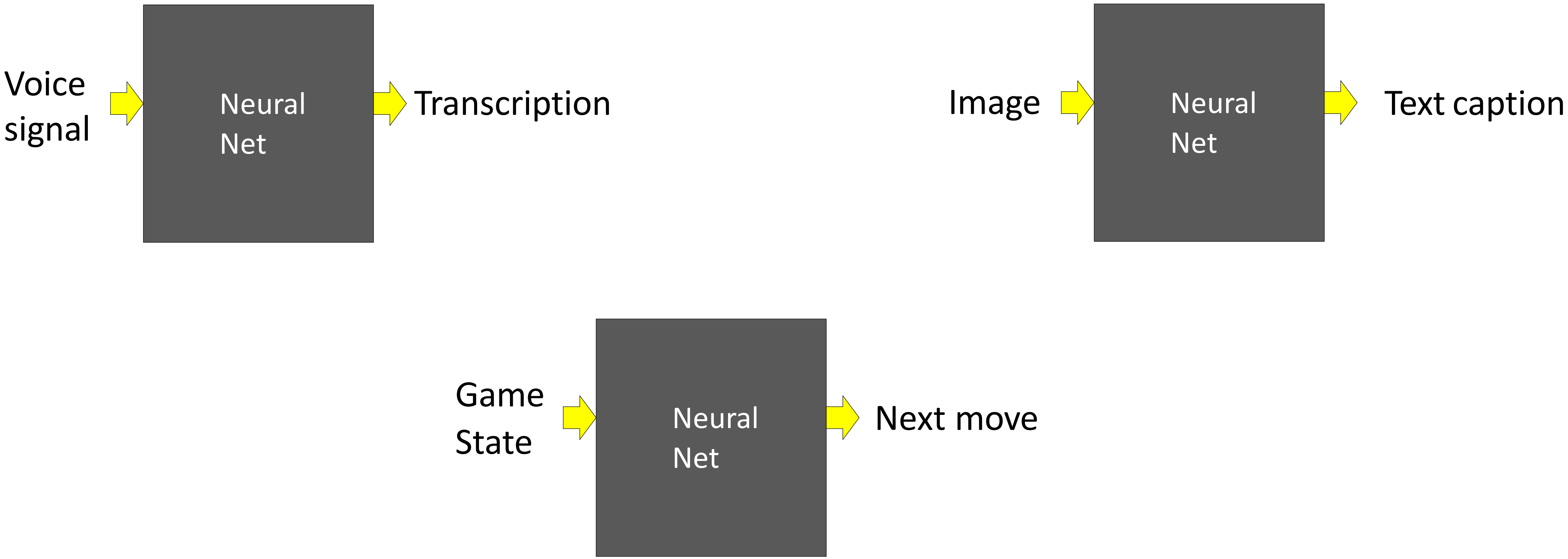
Deep models are taking over!

- Deep models have become one of *the* main approaches to AI

- They have been successfully applied to various pattern recognition, prediction, and analysis problems

- In many problems they have established the state of the art
  - Often exceeding previous benchmarks by large margins
  - Sometimes they solve problems that couldn’t be solved using earlier ML methods

So, what are deep models?



What's in these boxes?

The magical capacity of humans

- Humans can
  - Learn
  - Solve problems
  - Recognize patterns
  - Create
  - Cogitate
  - …

- Worthy of emulation
- But how do humans “work“?

Cognition and the brain..

Early Models of Human Cognition

- Associationism
  – Humans learn through association
- 400BC-1900AD
  -- Plato, David Hume, Ivan Pavlov..

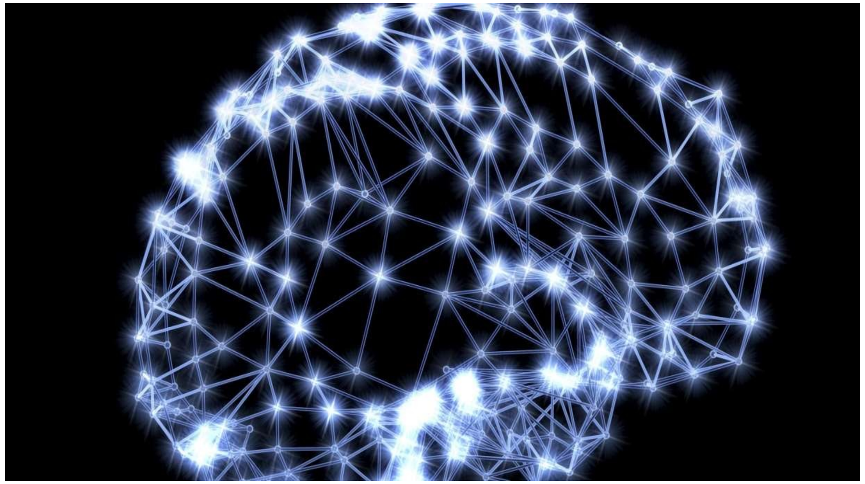
- Lightning is generally followed by thunder
  – “hey here’s a bolt of lightning, we’re going to hear thunder”
  – “We just heard thunder; did someone get hit by lightning”?

- **But where are the associations stored?**
- **And how? -** David Hartley’s *Observations on man* (1749)

Dawn of Connectionism

Observation: The Brain

Mid 1800s: The brain is a mass of interconnected neurons



- Many neurons connect *in* to each neuron
- Each neuron connects *out* to many neurons
- The brain is a *network* of neurons

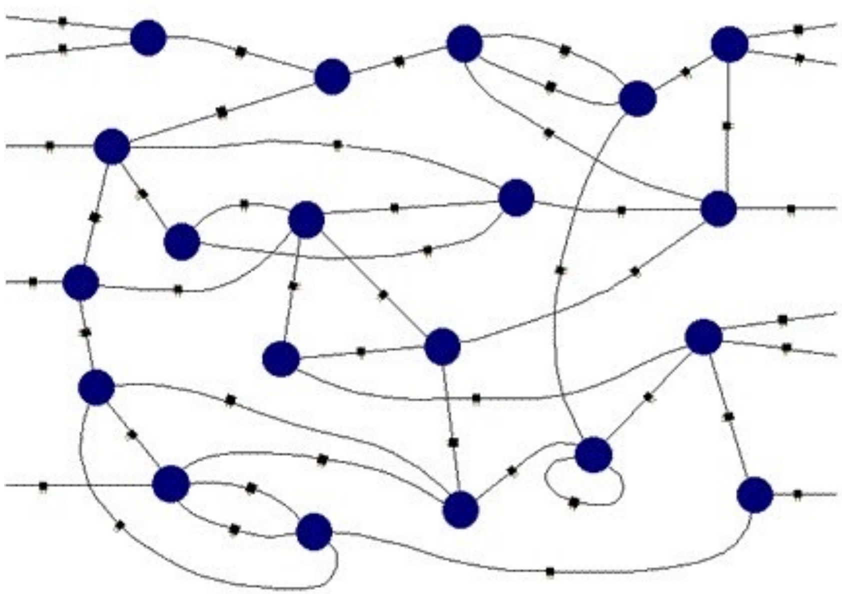
Enter *Connectionism*

- Alexander Bain, philosopher, psychologist, mathematician, logician, linguist, professor
- 1873: The information is in the ***connections***
  - *Mind and body* (1873)

- The human brain is a connectionist machine
- Neurons connect to other neurons. The processing/capacity of the brain is a function of these connections

- Connectionist machines emulate this structure

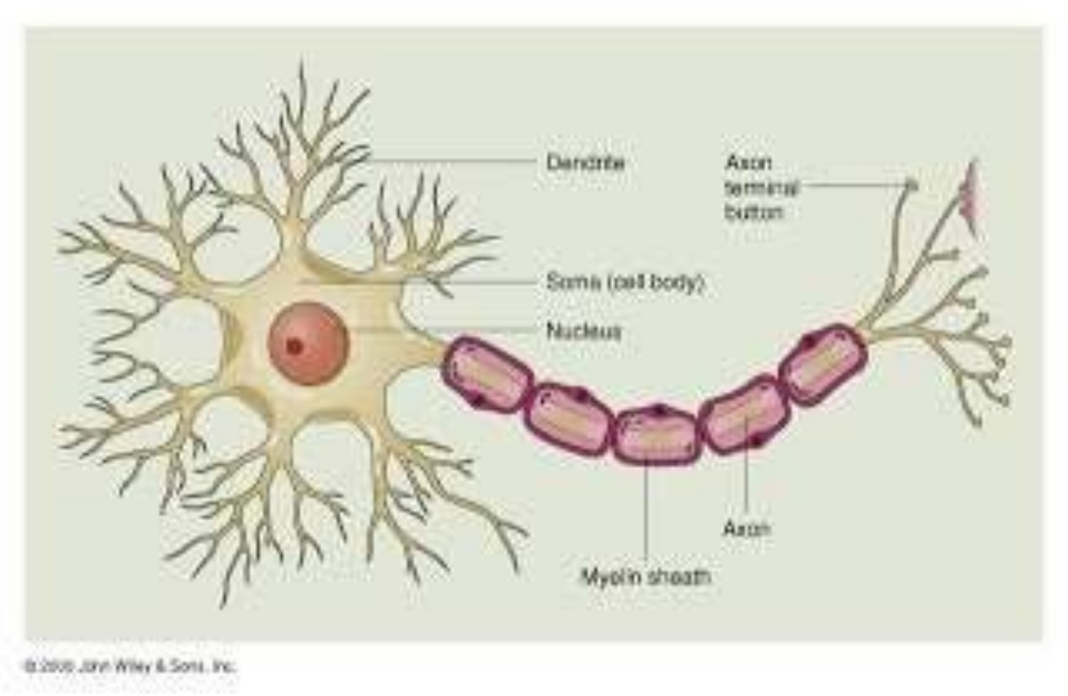
Connectionist Machines



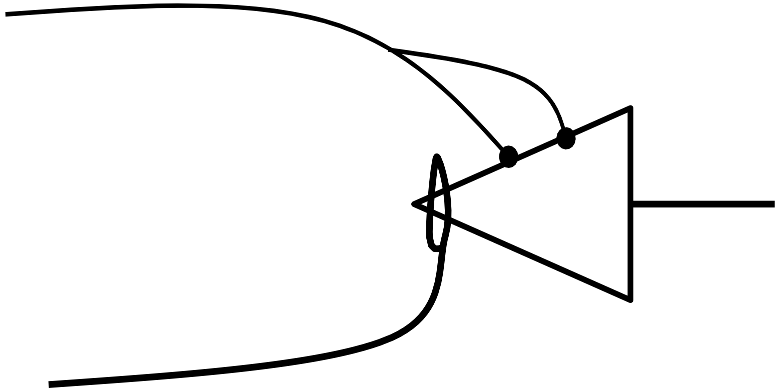
- Network of processing elements
  - **All world knowledge is stored in the *connections* between the elements**
- Neural networks are *connectionist* machines

Modelling the brain

- *But what are the individual elements?*
- A neuron:



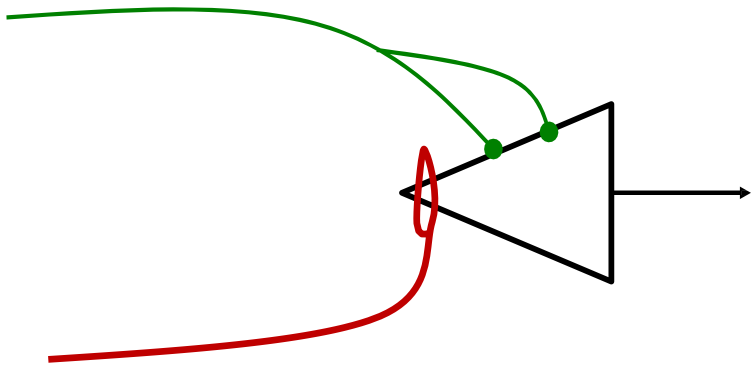
- A mathematical model of a neuron



Synaptic Model

  – McCulloch, W.S. & Pitts, W.H. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics, 5:115-137, 1943
    - Pitts was only 20 years old at this time

McCulloch and Pitts: Synaptic Model



- ***Excitatory synapse:*** Transmits weighted input to the neuron
- ***Inhibitory synapse:*** Any signal from an inhibitory synapse prevents neuron from firing
  - The activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time.
    - Regardless of other inputs

- Could compute arbitrary Boolean propositions
  - Since any Boolean function can be emulated, any Boolean function can be composed
- Models for *memory*
  - Networks with loops can "remember"
  - Lawrence Kubie (1930): Closed loops in the central nervous system explain memory
- Criticisms

  - Didn't provide a learning mechanism..

Donald Hebb

Novelist, farmer, hobo,
schoolteacher psychologist

- "Organization of behavior", 1949
- A learning mechanism:
  - "When an axon of cell *A* is near enough to excite a cell *B* and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that *A*'s efficiency, as one of the cells firing *B*, is increased."

As *A* repeatedly excites *B*, its *ability* to excite *B* improves

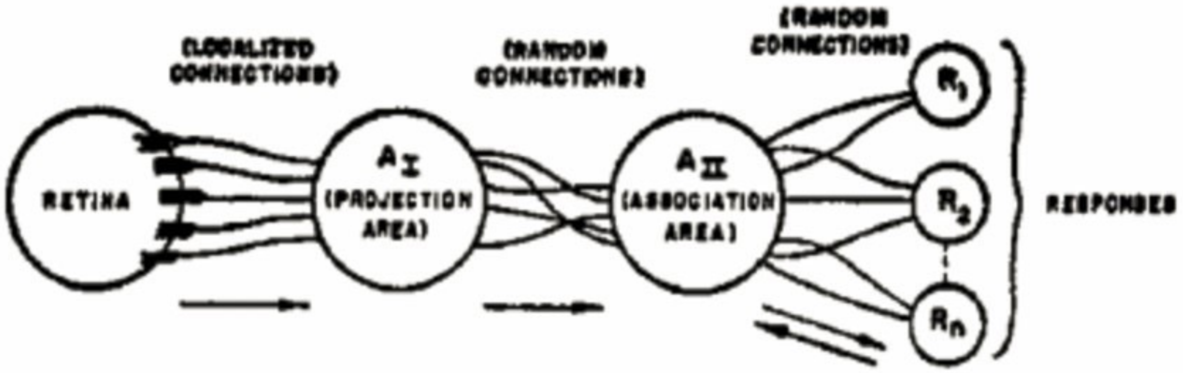
Hebbian Learning

- If neuron x repeatedly triggers neuron y, the synaptic knob connecting x to y gets larger
- In a mathematical model:

$$w_{xy} = w_{xy} + \eta xy$$

  – Weight of the connection from input neuron x to output neuron y
- This simple formula is actually the basis of many learning algorithms in ML
- Number of later modifications, allowing for weight normalization, forgetting etc.
  – E.g. Generalized Hebbian learning, aka Sanger’s rule

Perceptron

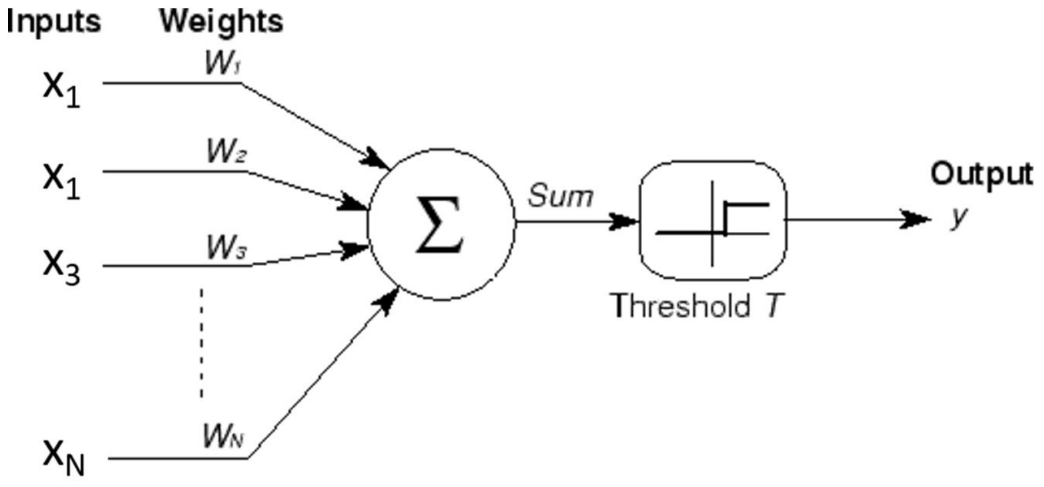


- Frank Rosenblatt
  – Psychologist, Logician
  – Inventor of the solution to everything, aka the Perceptron (1958)

- Original perceptron model
  – Groups of sensors (S) on retina combine onto cells in association area A1
  – Groups of A1 cells combine into Association cells A2
  – Signals from A2 cells combine into response cells R

– All connections may be excitatory or inhibitory

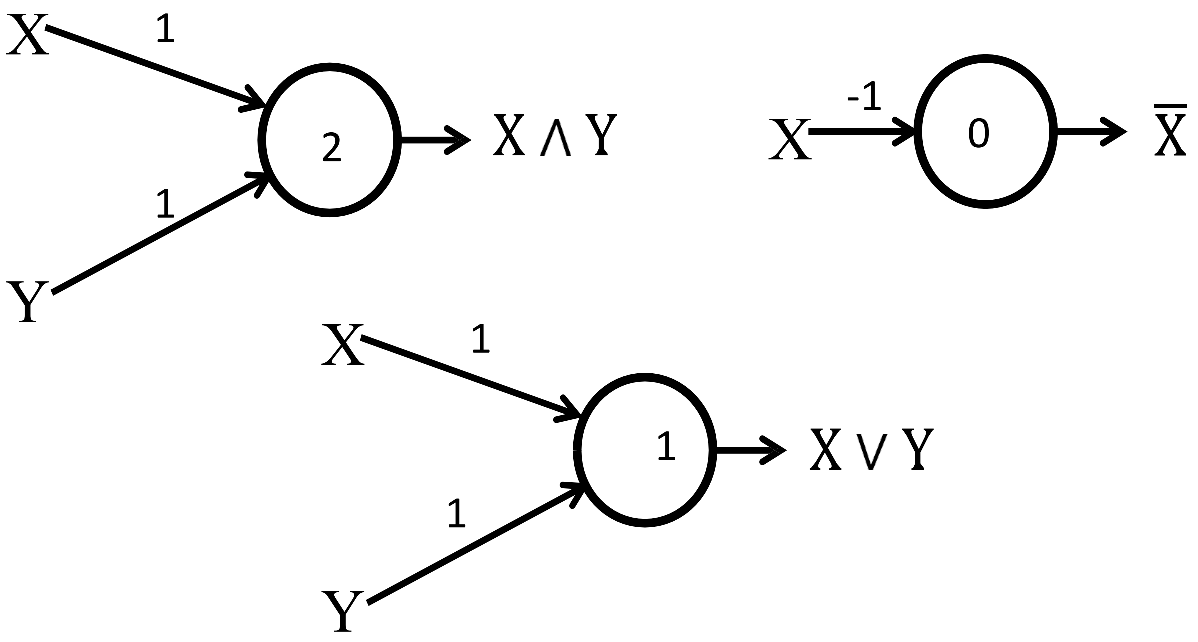
Perceptron: Simplified model



- Number of inputs combine linearly
  - Threshold logic: Fire if combined input exceeds threshold

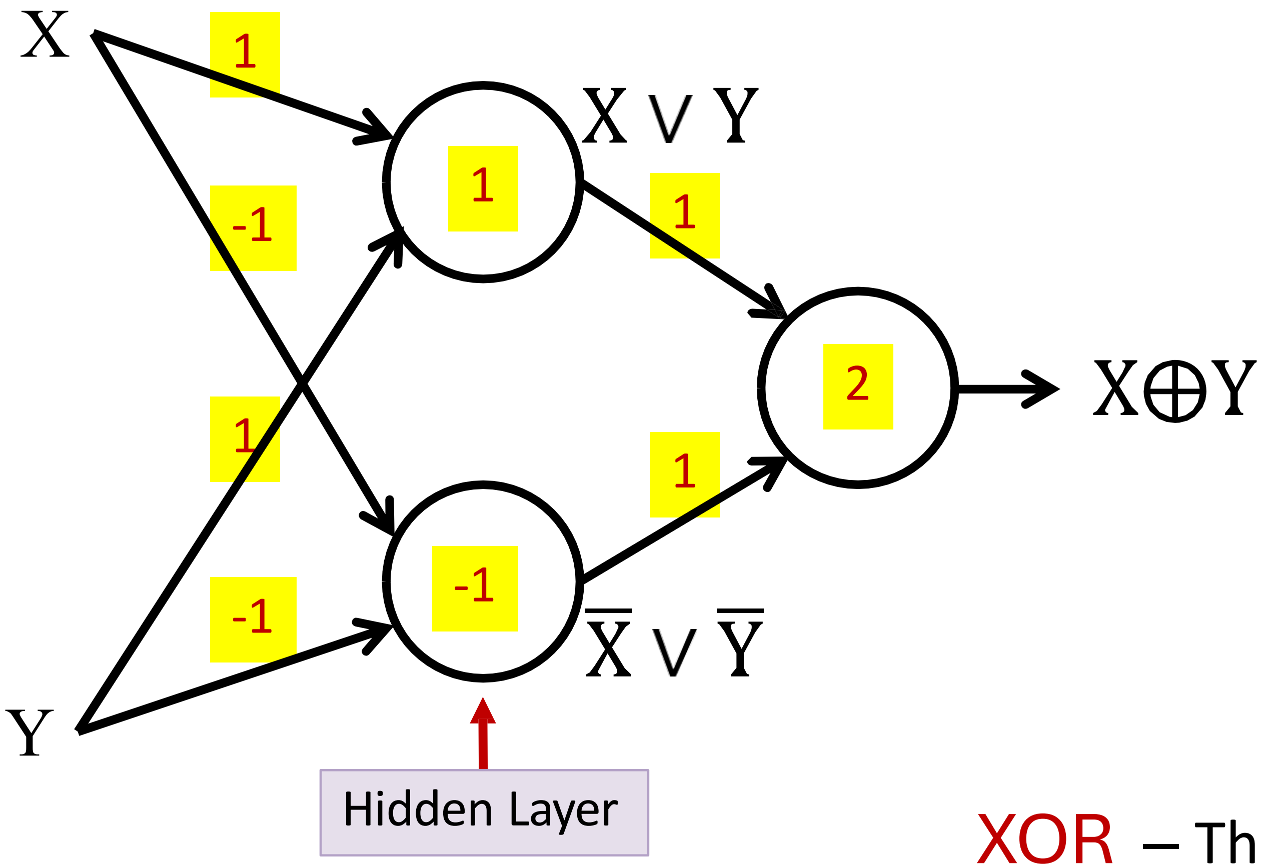
$$Y = \begin{cases} 1 \;\; if \sum_i w_i x_i - T \geq 0 \\ 0 \qquad\qquad\qquad else \end{cases}$$

Values shown on edges are weights, numbers in the circles are thresholds

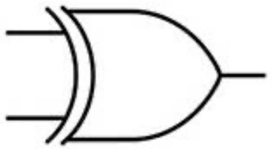


- Easily shown to mimic any Boolean gate
- No solution for XOR

Multi-layer Perceptron!



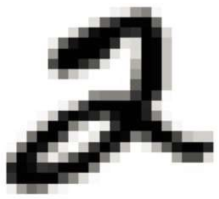
XOR



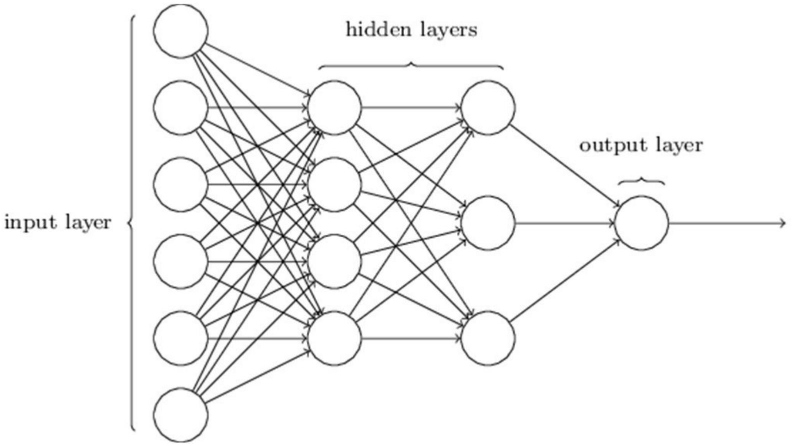
A	B	Output
0	0	0
1	0	1
0	1	1
1	1	0

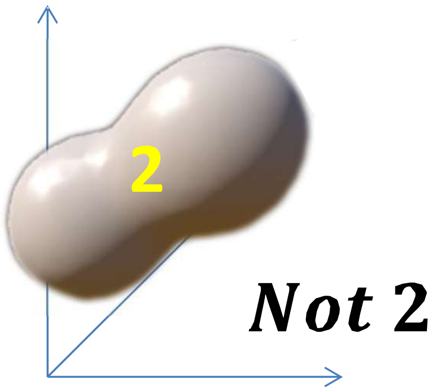
XOR – The first layer is a “hidden” layer

The MLP as a classifier



784 dimensions
(MNIST)





784 dimensions

- MLP as a function over real inputs
- MLP as a function that finds a complex "decision boundary" over a space of *reals*

Our brain is not Boolean

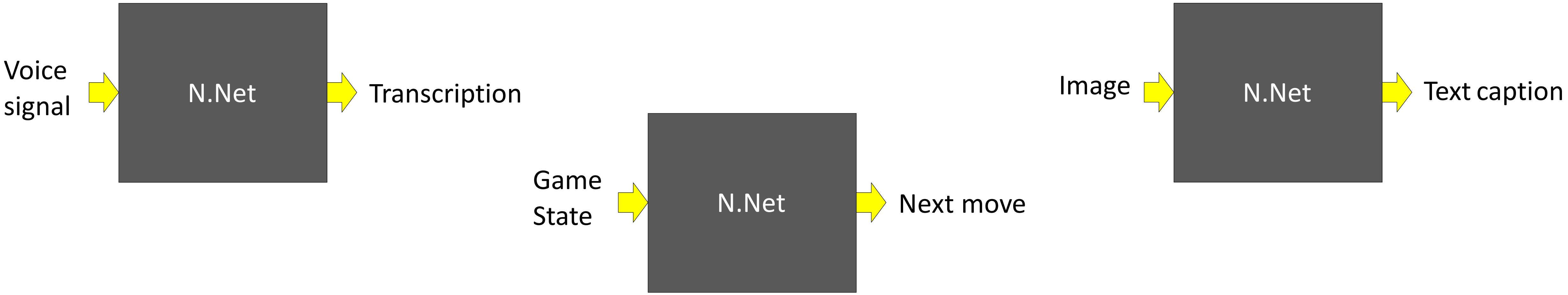
- We have real inputs
- We make non-Boolean inferences/predictions

MLP is a universal approximator

- MLPs can model any decision boundary
  - ✓ Non-Boolean input
- MLP can also model continuous valued functions
  - ✓ Non-Boolean output

Neural Nets in AI

- The neural network is a **function**
  - Given an input, it computes the function layer wise to predict an output
    - More generally, given one or more inputs, predicts one or more outputs



- Each of these boxes is actually a **function**
  - It can be approximated by a neural network

Landmarks in Deep Learning

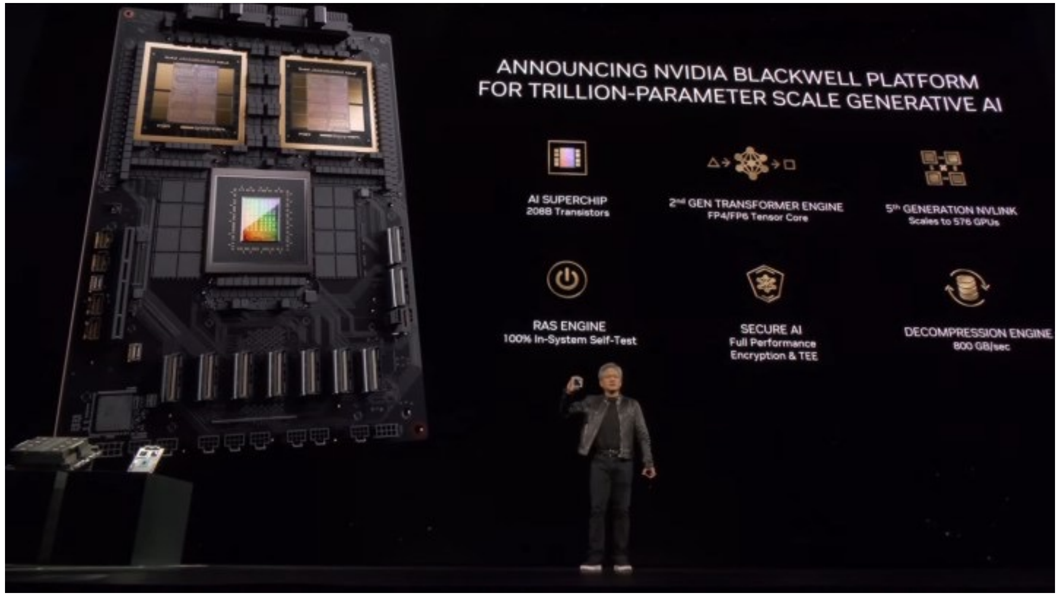
✓ 1958 Perceptron (Simple `neural' model)
- 1986 Backpropagation (Practical Deep Neural networks)
- 1989 Convolutional networks (Supervised learning)

✓ 2000 Graphics Processing Units (GPUs)

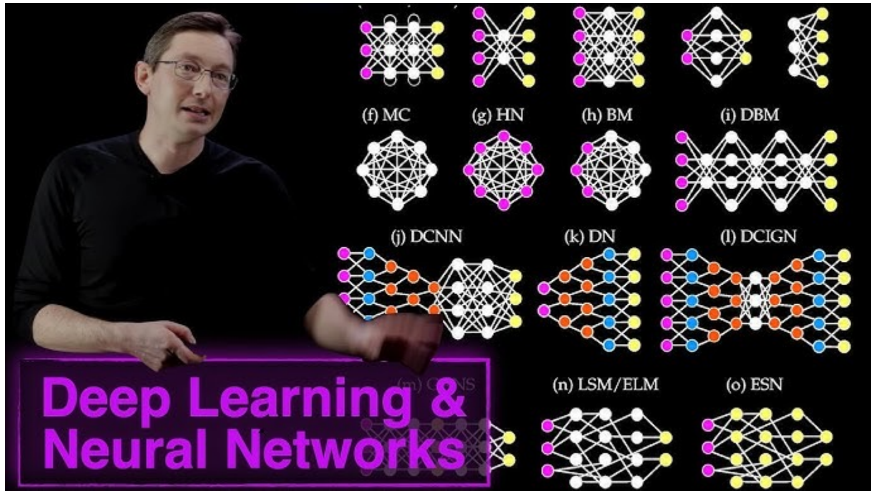
✓ 2009 ImageNet, a free database of more than 14 million labeled images

- 2012 AlexNet Image classification (Supervised learning)
- 2014 Generative adversarial networks (Unsupervised learning)
- 2014 Deep Q-Learning -- Atari games (Reinforcement learning)
- 2016 AlphaGo (Reinforcement learning)
- 2017 Machine translation (Supervised learning)
- 2019 Language models ((Un)supervised learning)
- 2022 Dall-E2 Image synthesis from text prompts ((Un)supervised learning)
- 2022 ChatGPT ((Un)supervised learning)
- 2023 GPT4 Multimodal model ((Un)supervised learning)

Why Deep Learning Now?



GPUS

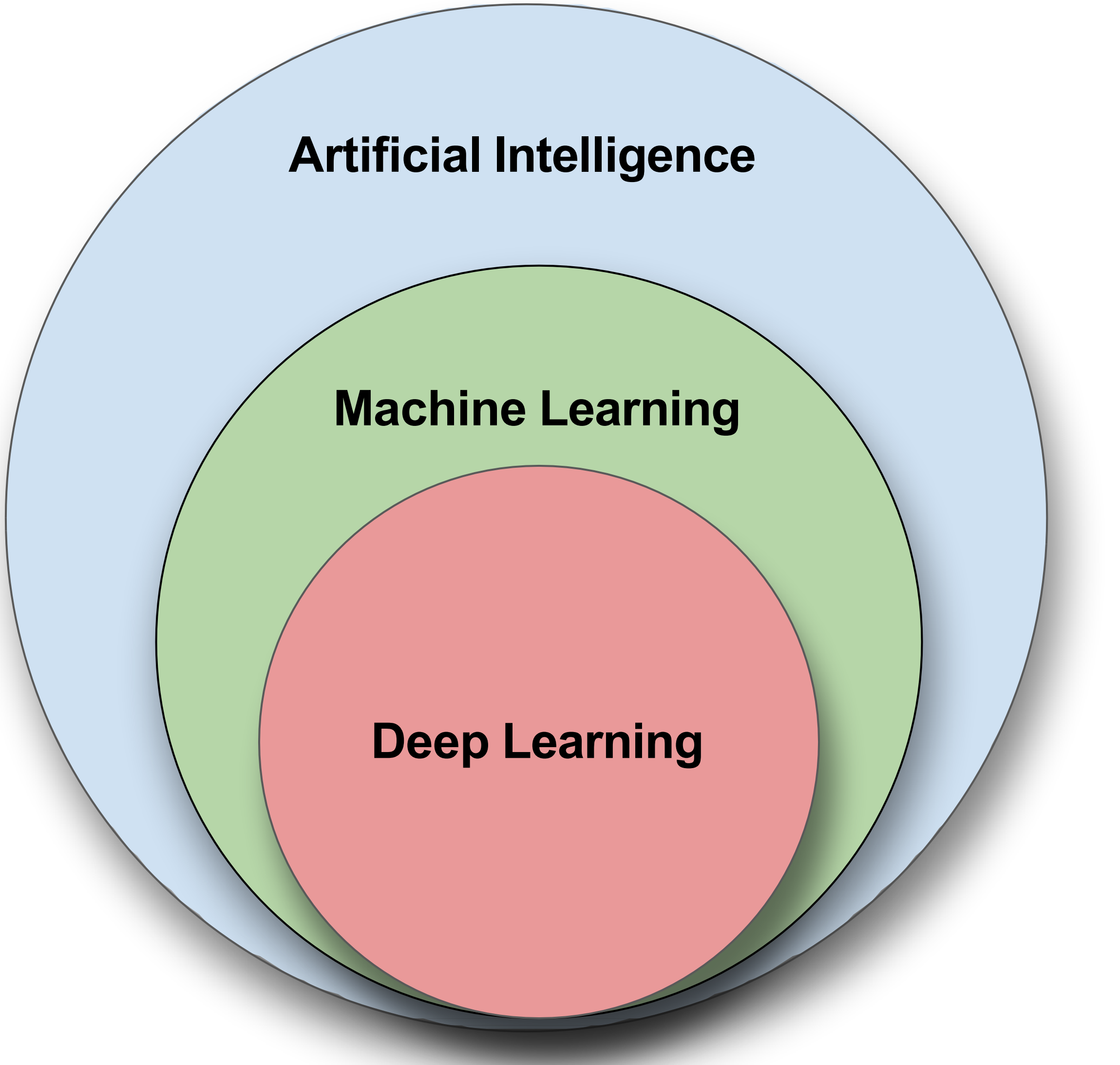


Algorithms

Big Data

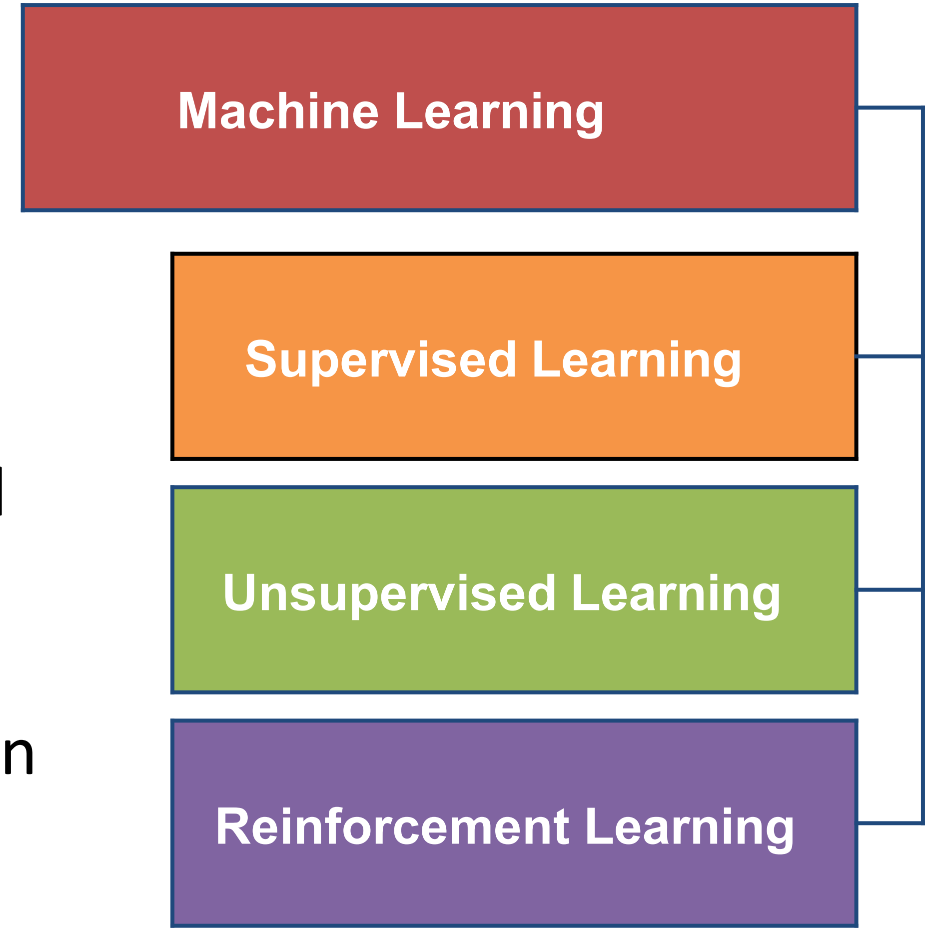
Relationship of AI, ML and DL

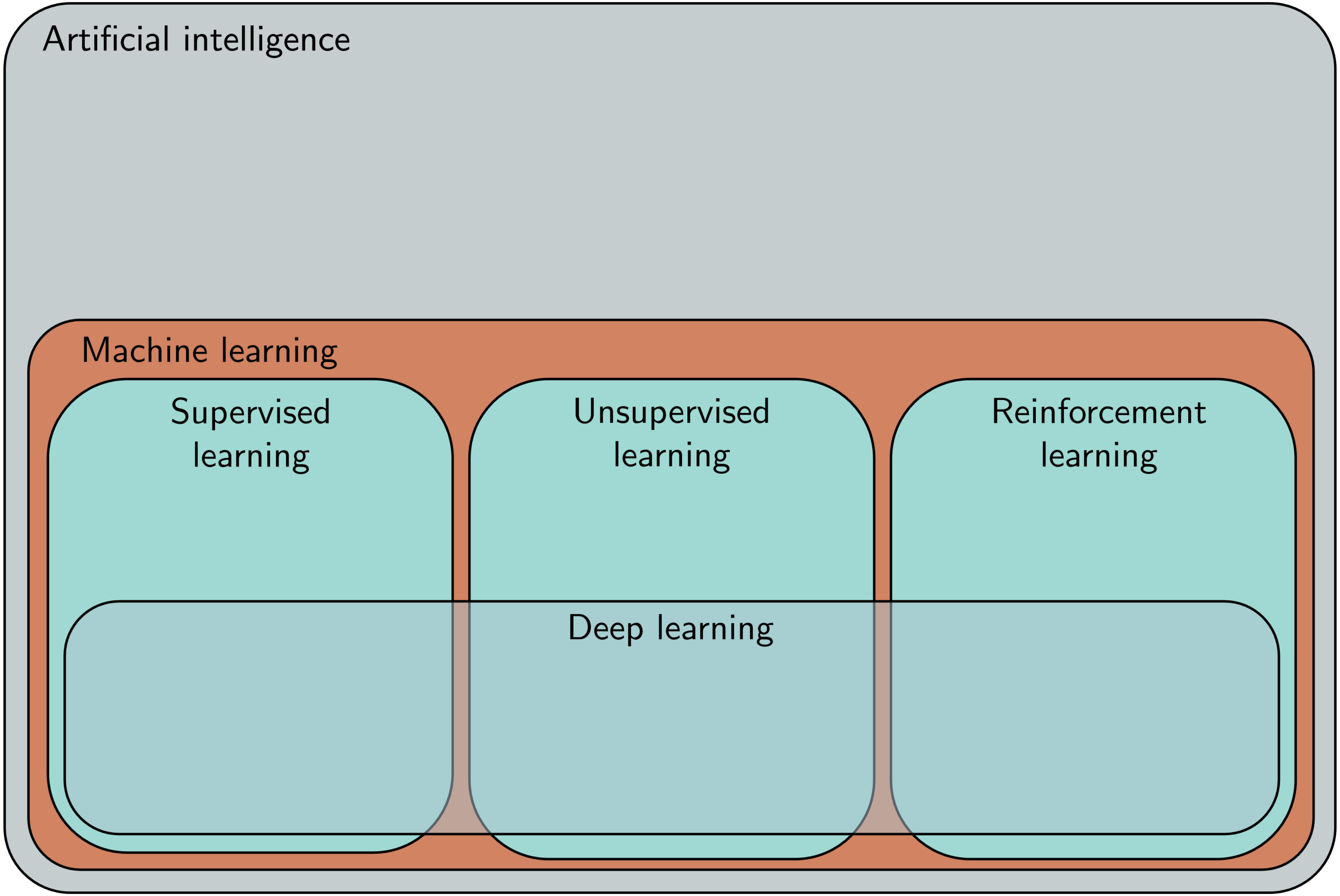
- **Artificial Intelligence (AI)** is anything about man-made intelligence exhibited by machines.
- **Machine Learning (ML)** is an approach to achieve **AI**.
- **Deep Learning (DL)** is one technique to implement **ML**.



Types of ML Algorithms

- Supervised Learning
  - trained with labeled data; including regression and classification problems
- Unsupervised Learning
  - trained with unlabeled data; clustering and association rule learning problems.
- Reinforcement Learning
  - no training data; stochastic Markov decision process; robotics and self-driving cars.

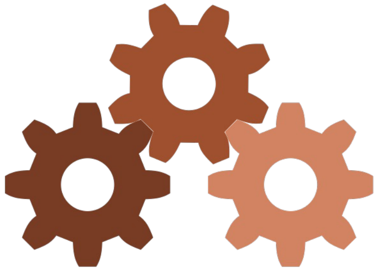



Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Supervised learning

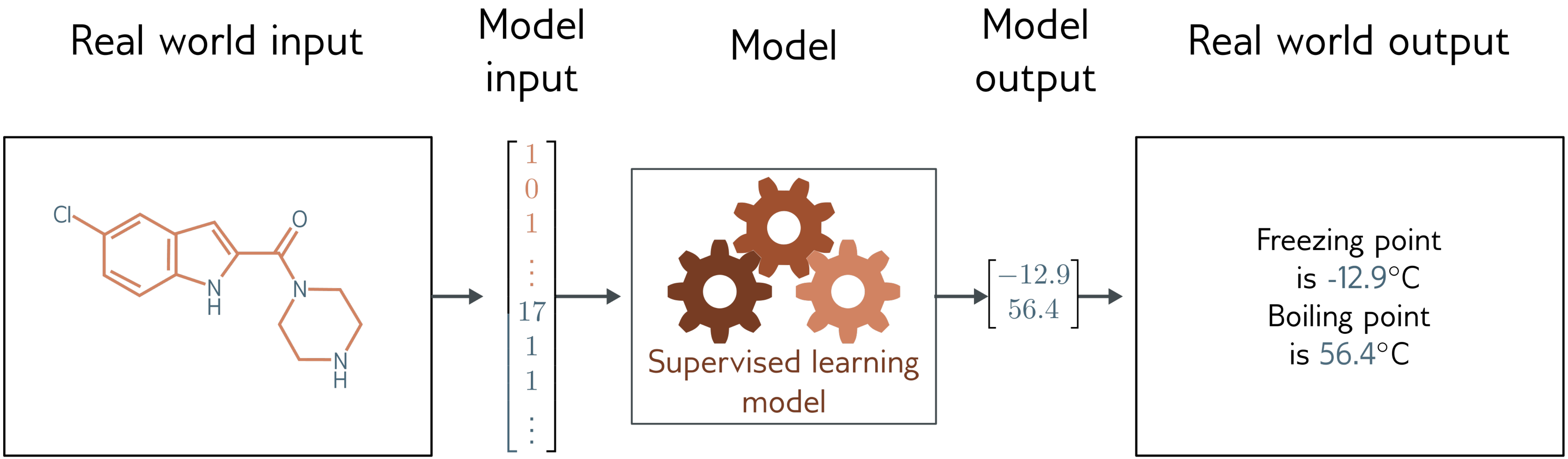
- Define a mapping from input to output
- Learn this mapping from paired input/output data examples

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.	$\begin{bmatrix} 6000 \\ 4 \\ 235 \\ 2005 \\ 1 \end{bmatrix}$	 Supervised learning model	$[340]$	Predicted price is $340k

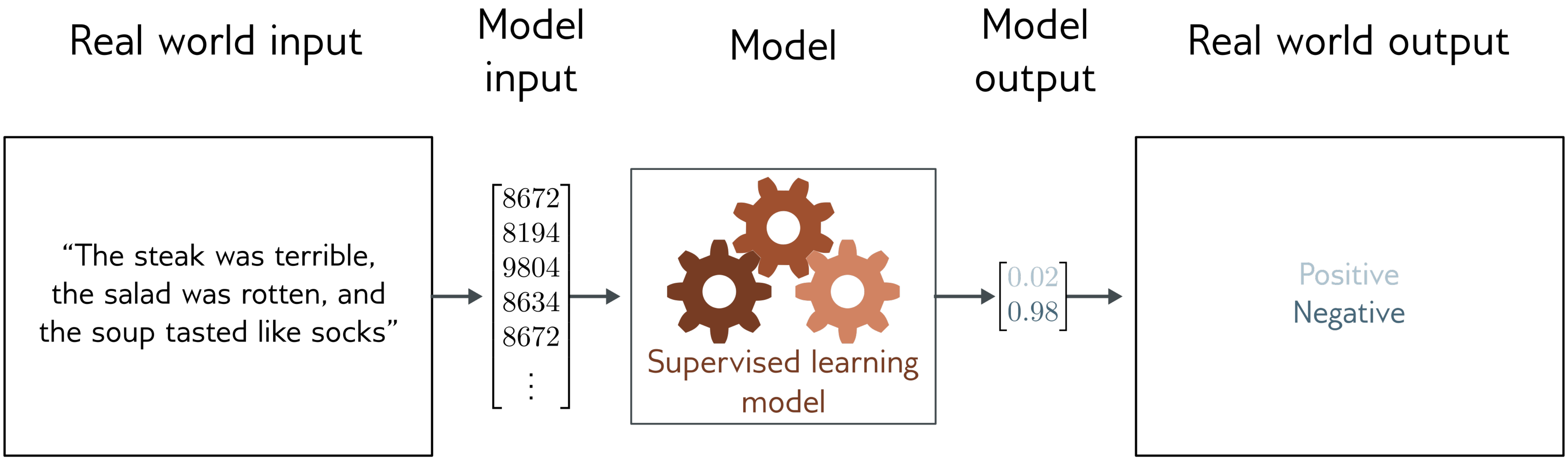
- Univariate regression problem (one output, real value)
- Fully connected network

Graph regression



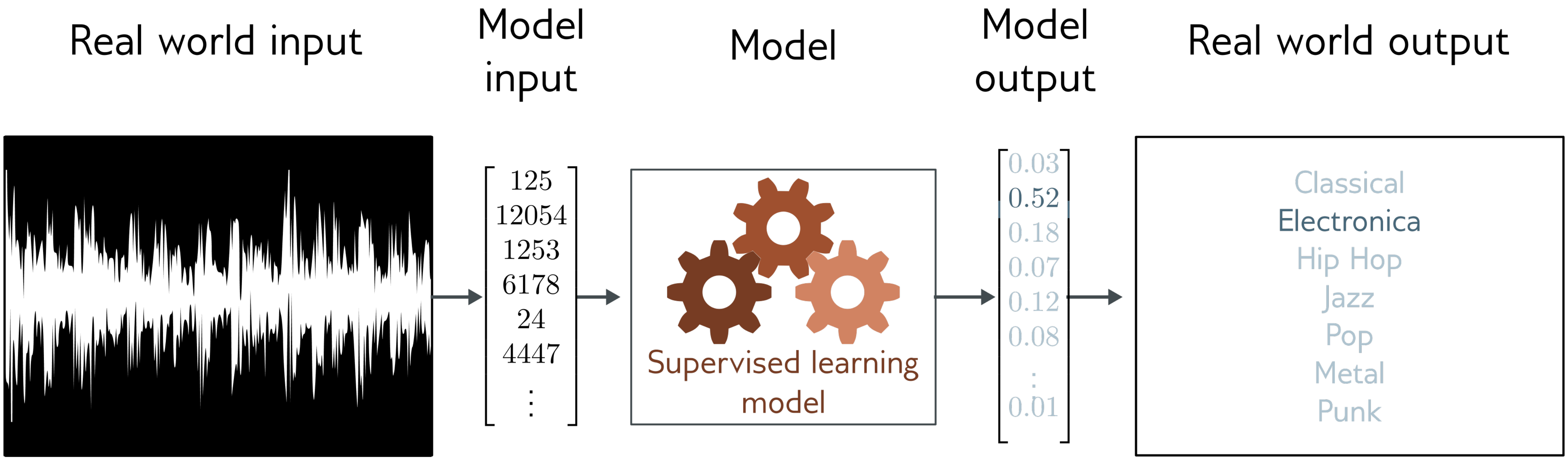
- Multivariate regression problem (>1 output, real value)
- Graph neural network

Text classification



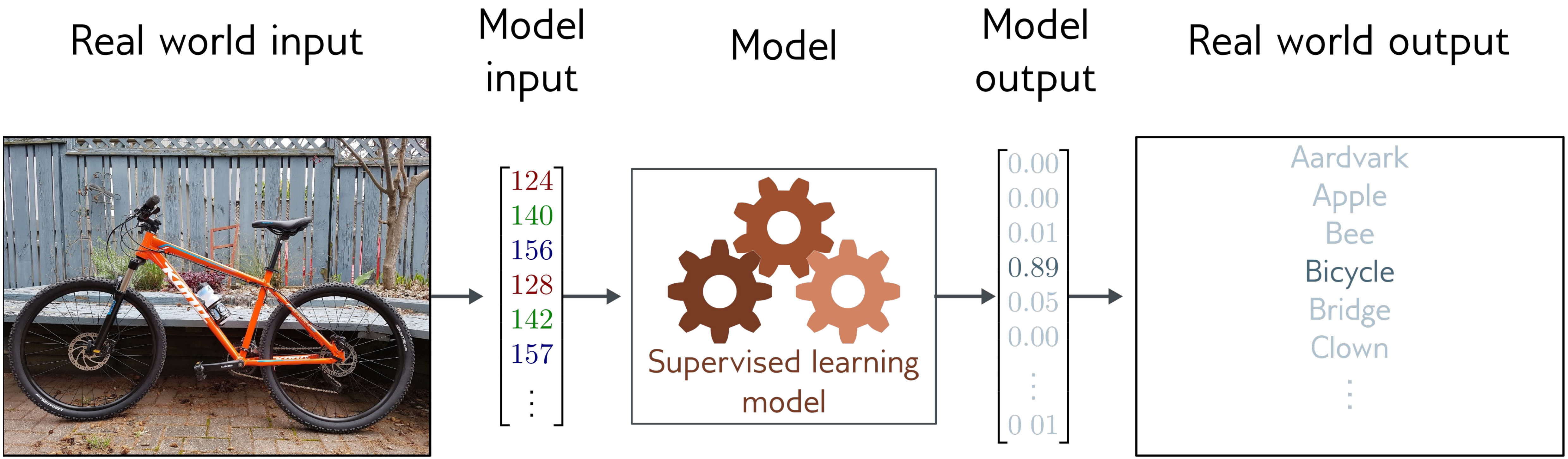
- Binary classification problem (two discrete classes)
- Transformer network

Music genre classification



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

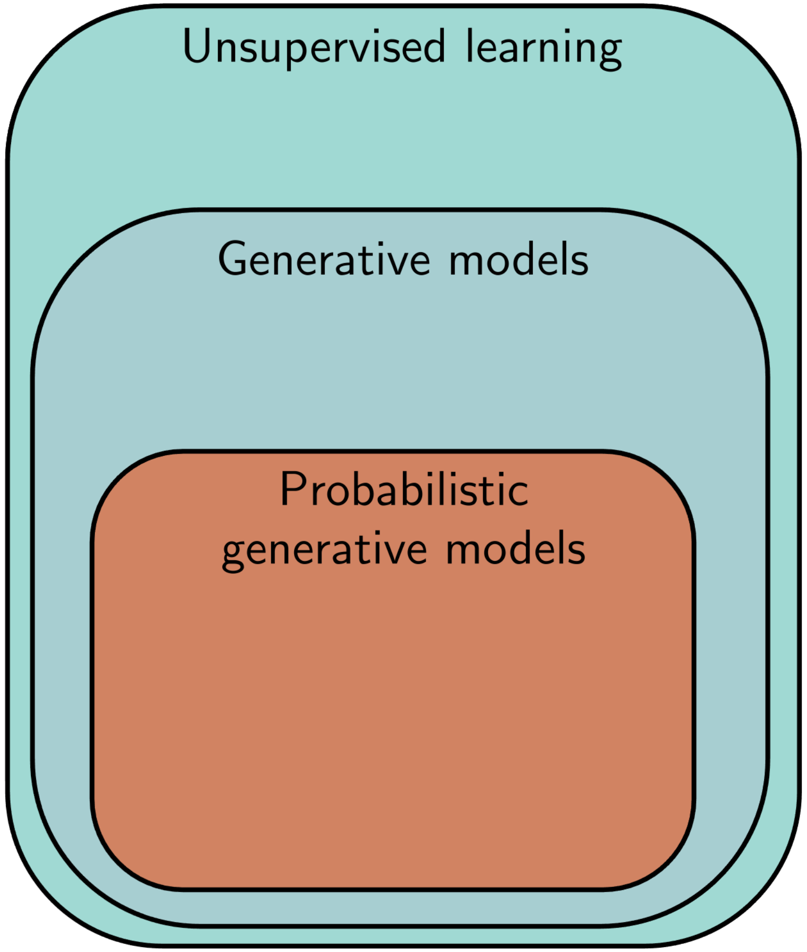
Image classification



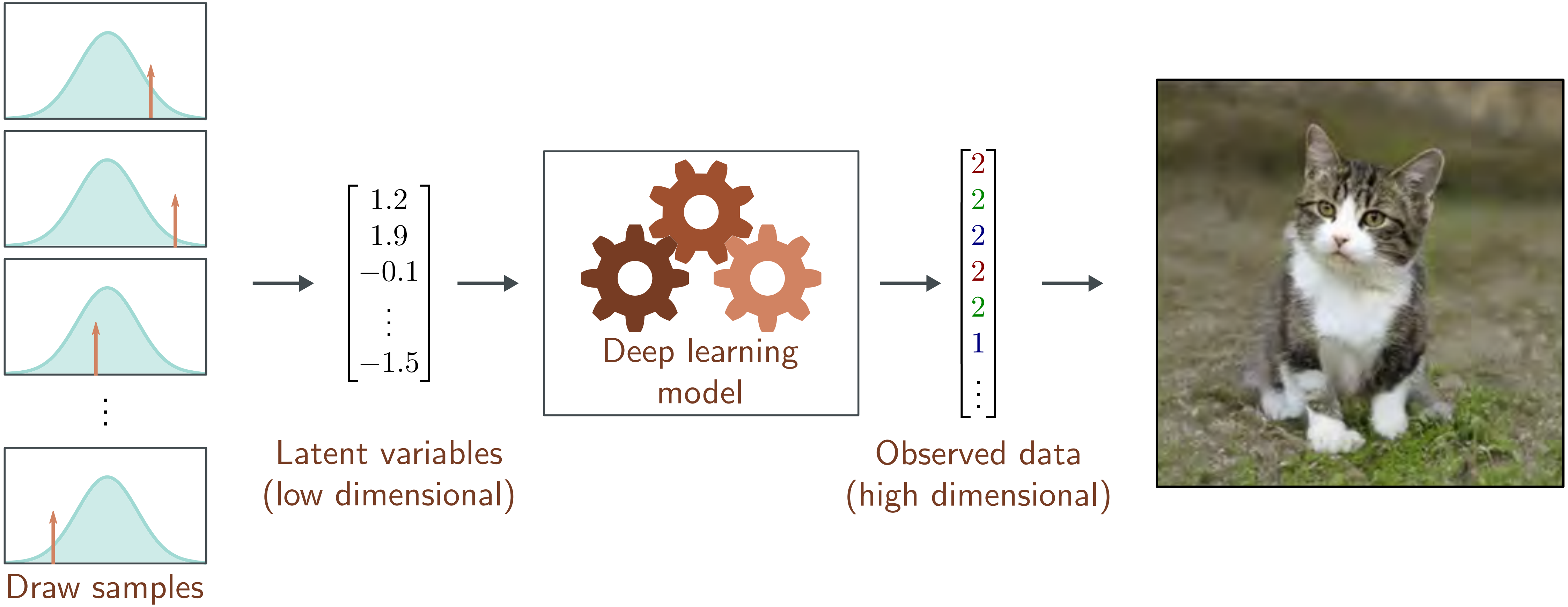
- Multiclass classification problem (discrete classes, >2 possible classes)
- Convolutional network

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



Latent variables



Conditional synthesis

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

- *We will not cover reinforcement learning in this class*

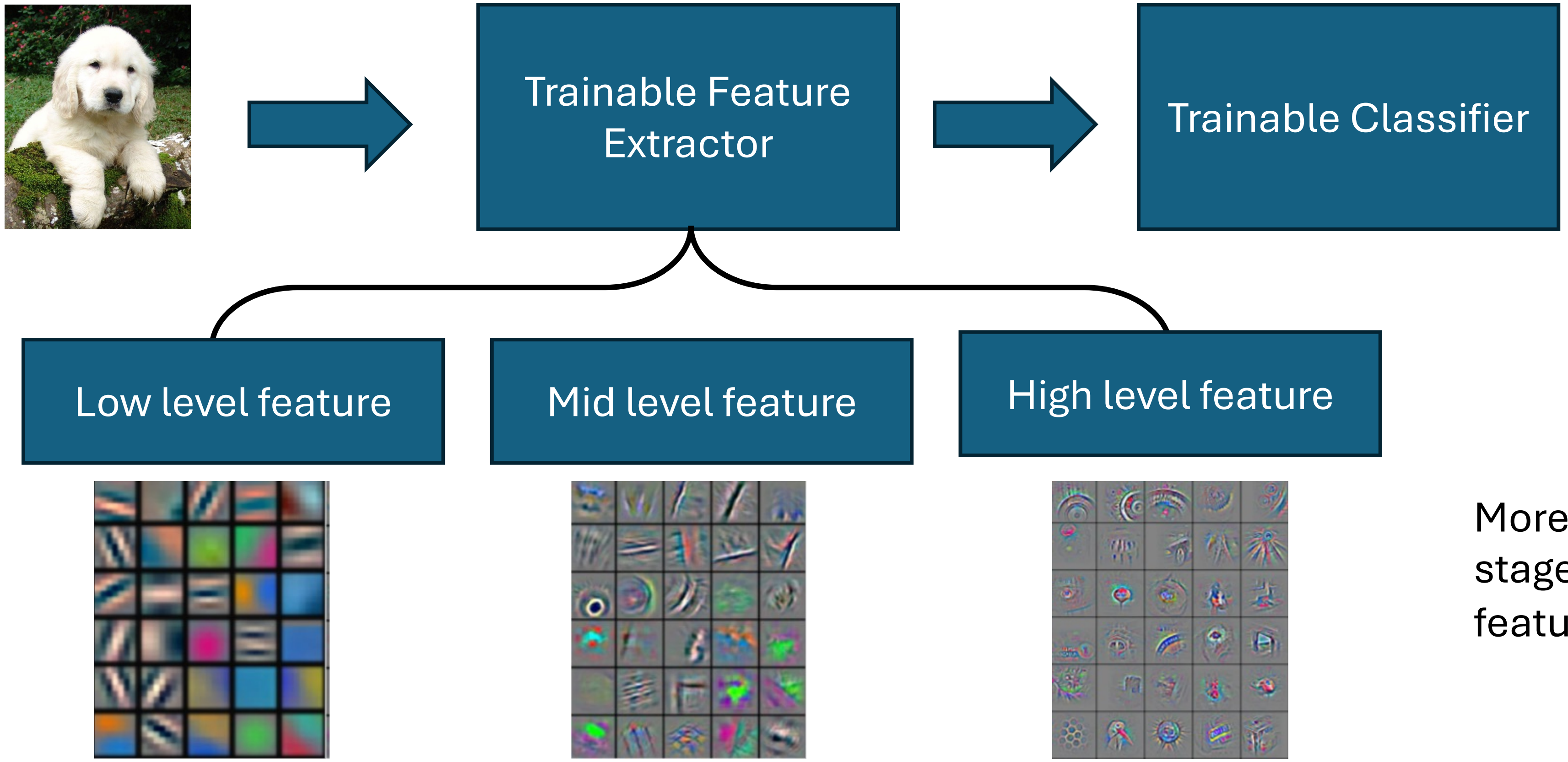
Difference between DL and traditional ML?

- Traditional model of pattern recognition: fixed/hand-engineered features + trainable classifier

- End-to-end Learning/feature learning/deep learning: trainable features + trainable classifier

Deep Learning = Learning Representations

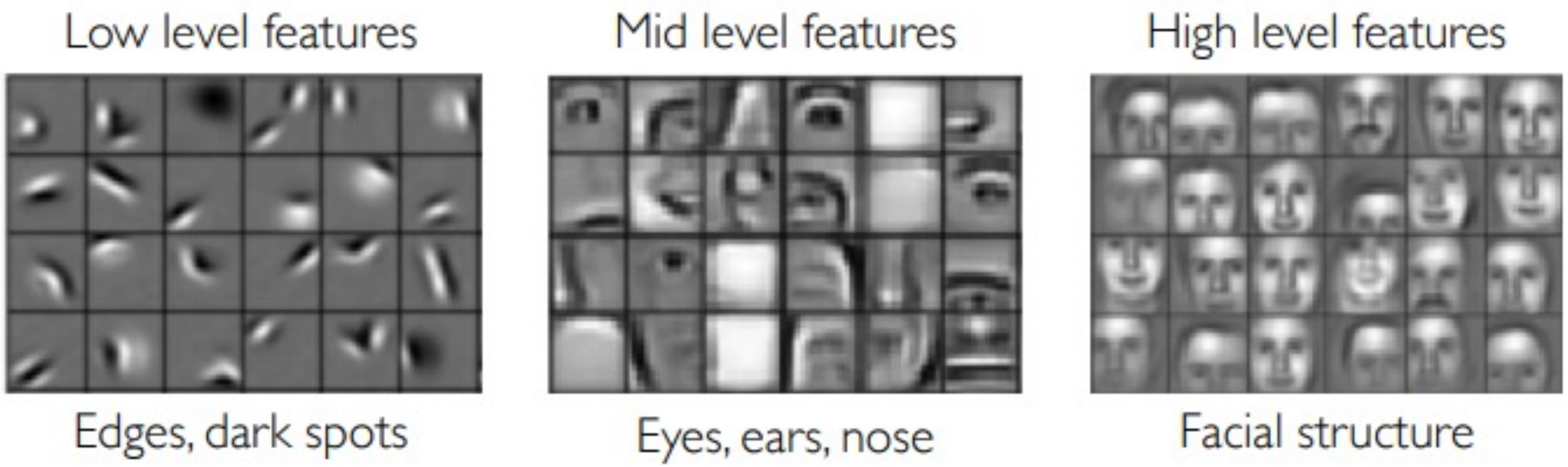
- Deep architecture: learn hierarchical representations



More than one stage of non-linear feature extraction

Trainable Feature Hierarchies

- A hierarchy of trainable feature transforms
  - Each module transforms its input representation into a higher-level representation
  - High-level features are more global and more invariant
  - Low-level features are shared among categories



- Deep learning Goal: make all modules trainable and get them to learn appropriate representations

Class Structure Part 1
(subject to change as class progresses)

- Basic concepts about deep learning

Week of	**Tuesday**	**Thursday**
1/13	1 Introduction	2 Why Deep
1/20	3 Learning a Neural Network	4 Forward and Backward Propagation
1/27	5 Automatic Differentiation	6 Create Your Model with PyTorch
2/3	7 Train Your Model: Optimization 1 (Convergence and Learning Rate)	8 Train Your Model: Optimization 2 (Stochastic GD and Mini Batch)
2/10	9 Tame Your Model: Tricks	Classes Not Held

Class Structure Part 2

(subject to change as class progresses)

- Popular architectures with different data types

Week of	**Tuesday**	**Thursday**
2/17	10 Convolutional Neural Network	11 Convolutional Neural Networks - 2
2/24	12 Recurrent Neural Network	13 LSTM
3/3	No Class (Spring Break)	No Class (Spring Break)
3/10 * **Last day to drop classes**	14 Seq-to-Seq	15 Attention
3/17	16 Transformer	17 Transformer and LLMs
3/24	18 Graph Neural Network	19 Graph Neural Network 2

Class Structure Part 3
(subject to change as class progresses)

- Probabilistic deep learning: generative learning and beyond

Week of	**Tuesday**	**Thursday**
3/31	20 Probabilistic Deep Learning: Introduction	21 Generative Modeling
4/7	22 Generative Modeling 2	23 Diffusion Models
4/14	24 Diffusion Models 2	25 Bayesian Deep Learning
4/21	26 Deep Probabilistic Graphical Models	TBD