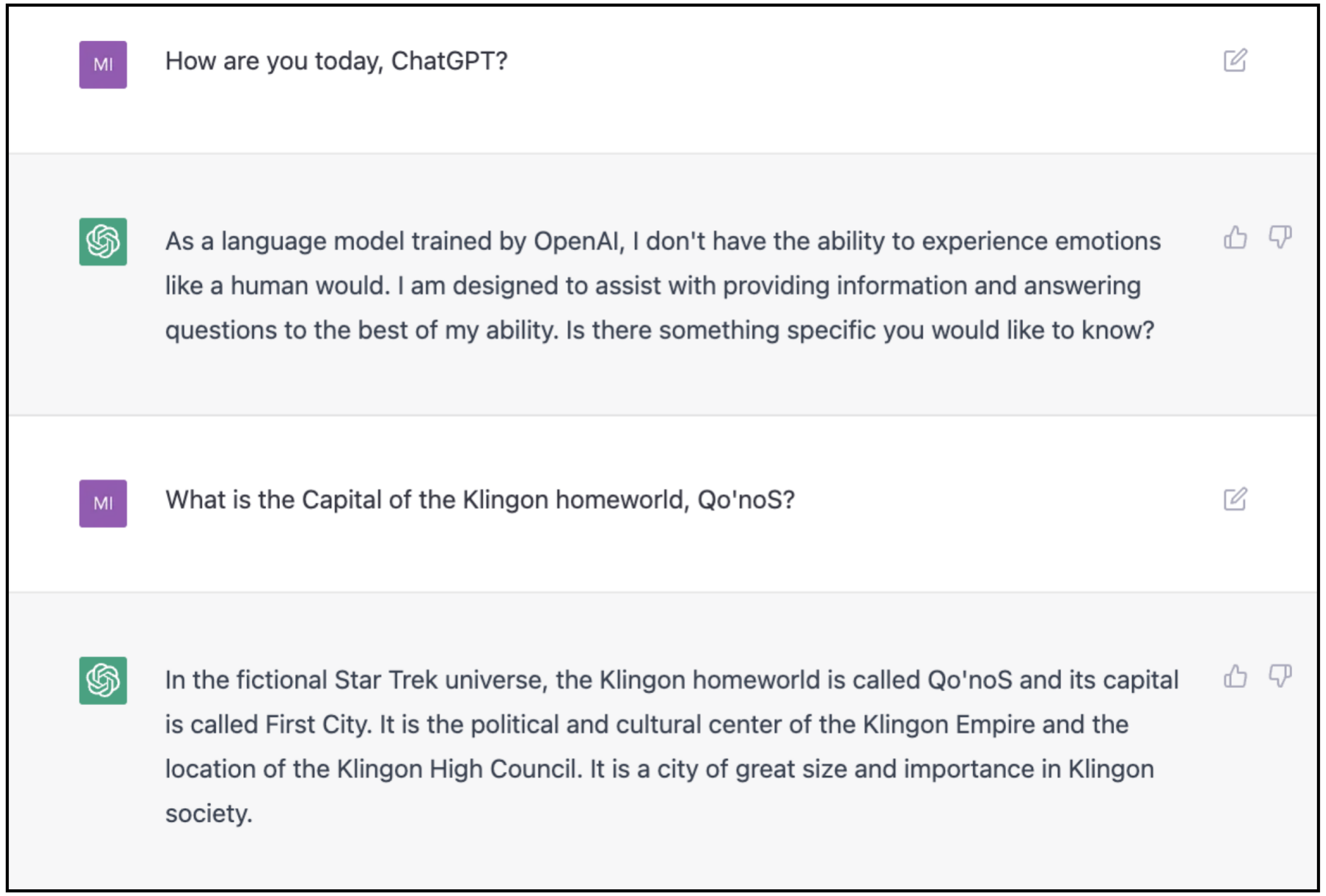
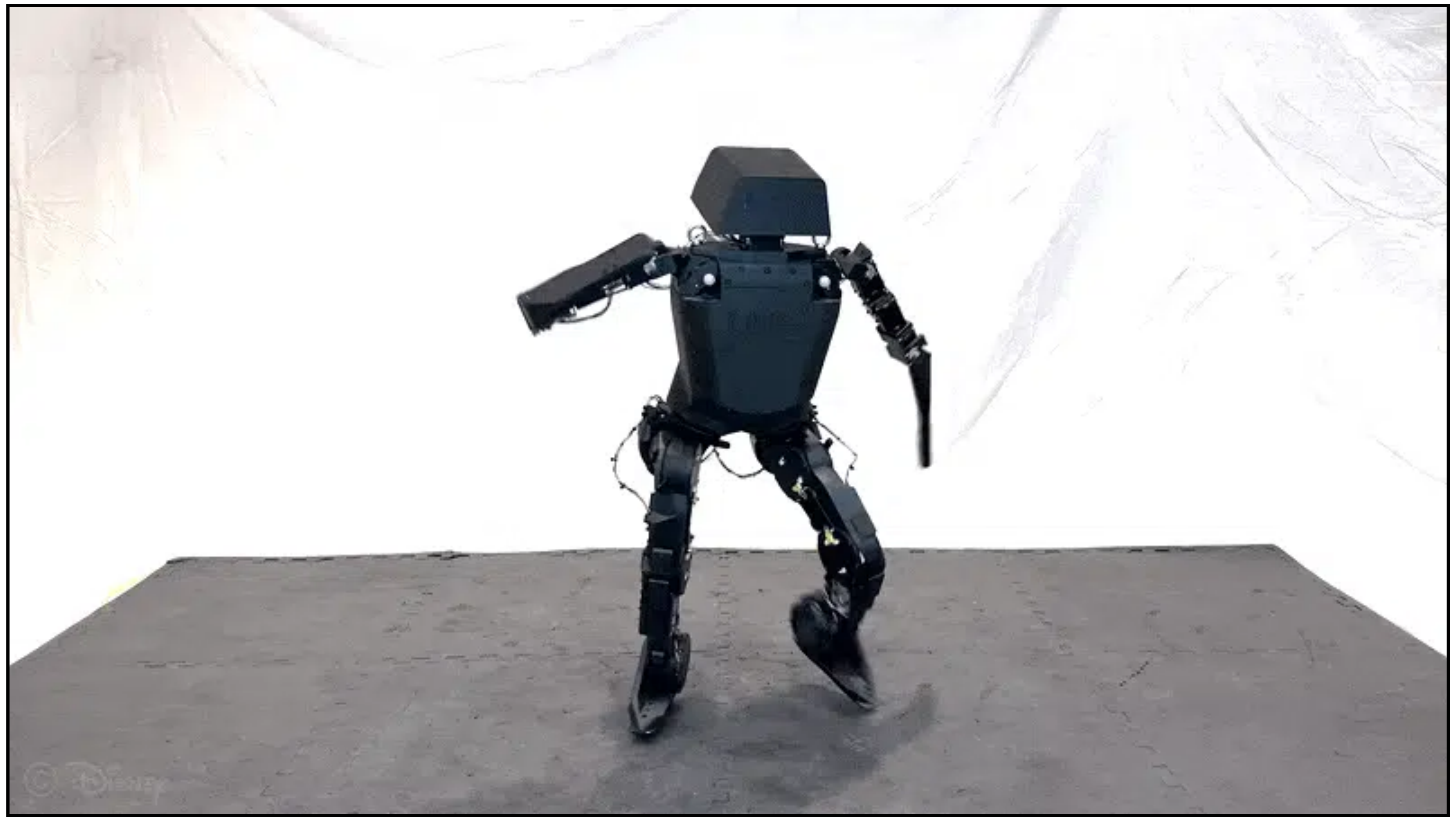
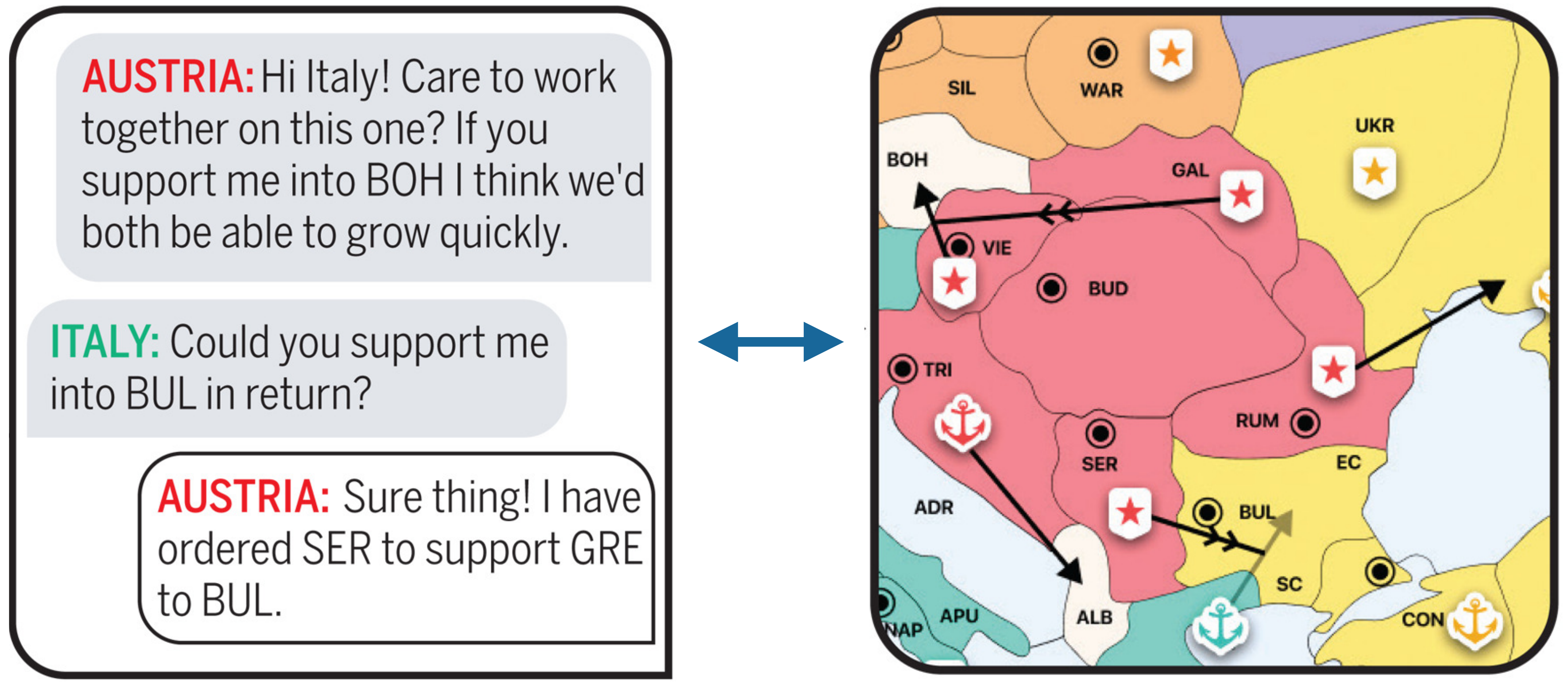
Introduction to machine learning



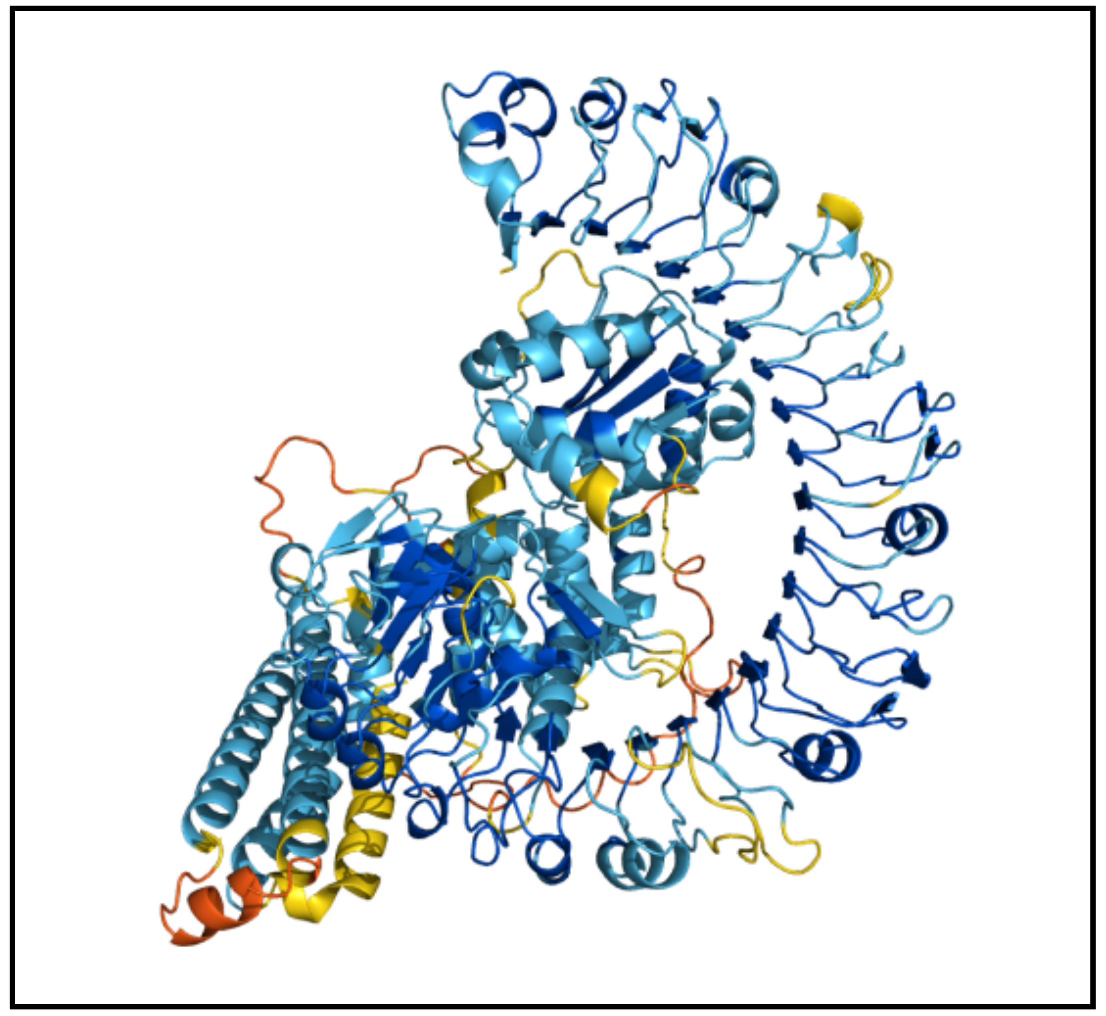
ChatGPT


Disney Research



Cicero


DALLE-2


AlphaFold

Roadmap

- What is machine learning?
- Machine learning paradigms
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Perceptron
  - Example: Golden retriever vs. Doberman
  - Perceptron model
  - Training via update rule
  - Testing
- Multi-layer Perceptron (MLP)
- Multi-class Perceptron

Reminder: Policy on generative AI tools in COS 126

Executive summary.

- Do not use generative AI tools for any purpose for graded assessments (unless explicitly allowed).

Reasons not to use generative AI (unless permitted):

- Enhance your critical thinking and problem-solving skills.
- Establish a strong foundation in programming (syntax, semantics, design, debugging, data structures, algorithms, performance, and theoretical principles).
- Will use generative AI more effectively in the future by learning to critically evaluate its output.
- Relying on AI-generated code without understanding how it works can lead to significant technical and ethical issues.
- We routinely detect the use of generative AI on assignments and refer these cases to the CoD.

Full details. See course syllabus.

ML data ⇔ related fields

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9

Tabular

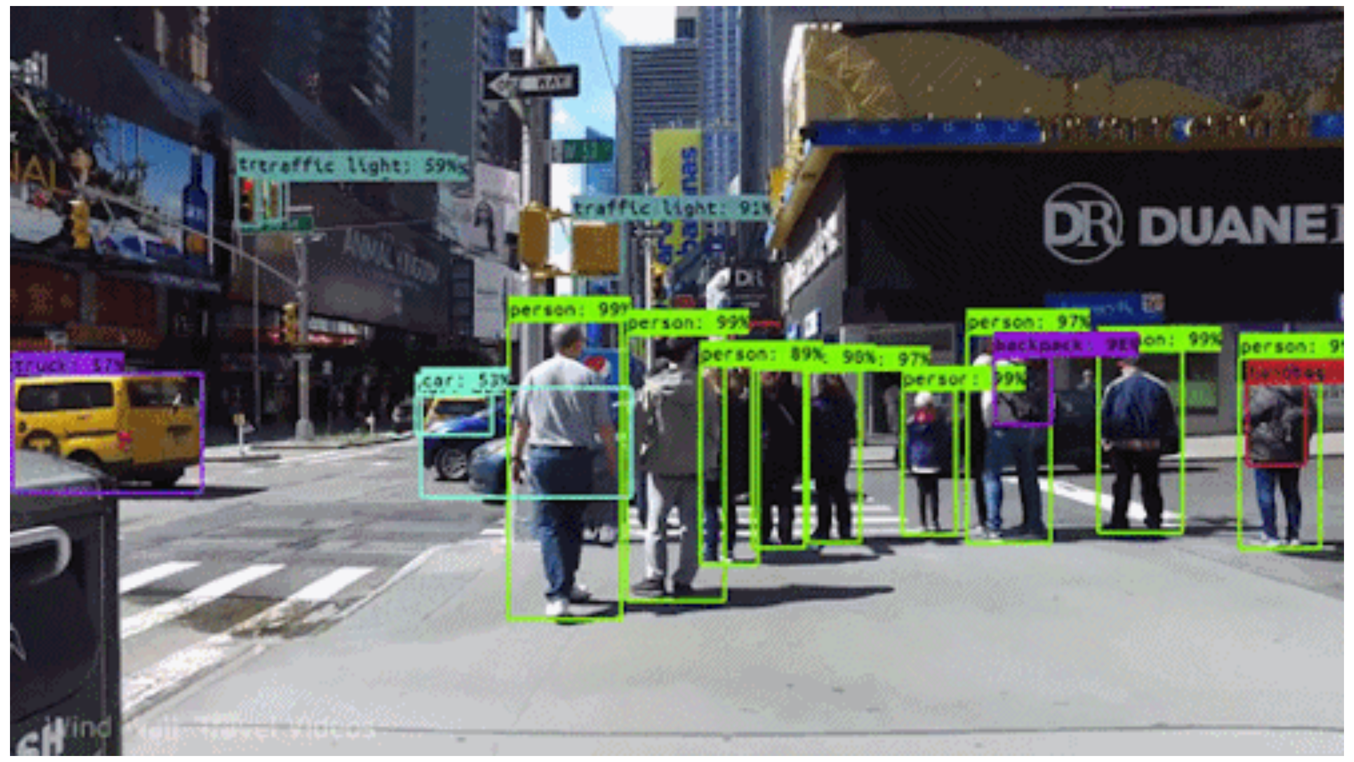
Visual

ML data ⇔ related fields

Player	Minutes	Points	Rebounds	Assists
A	41	20	6	5
B	30	29	7	6
C	22	7	7	2
D	26	3	3	9

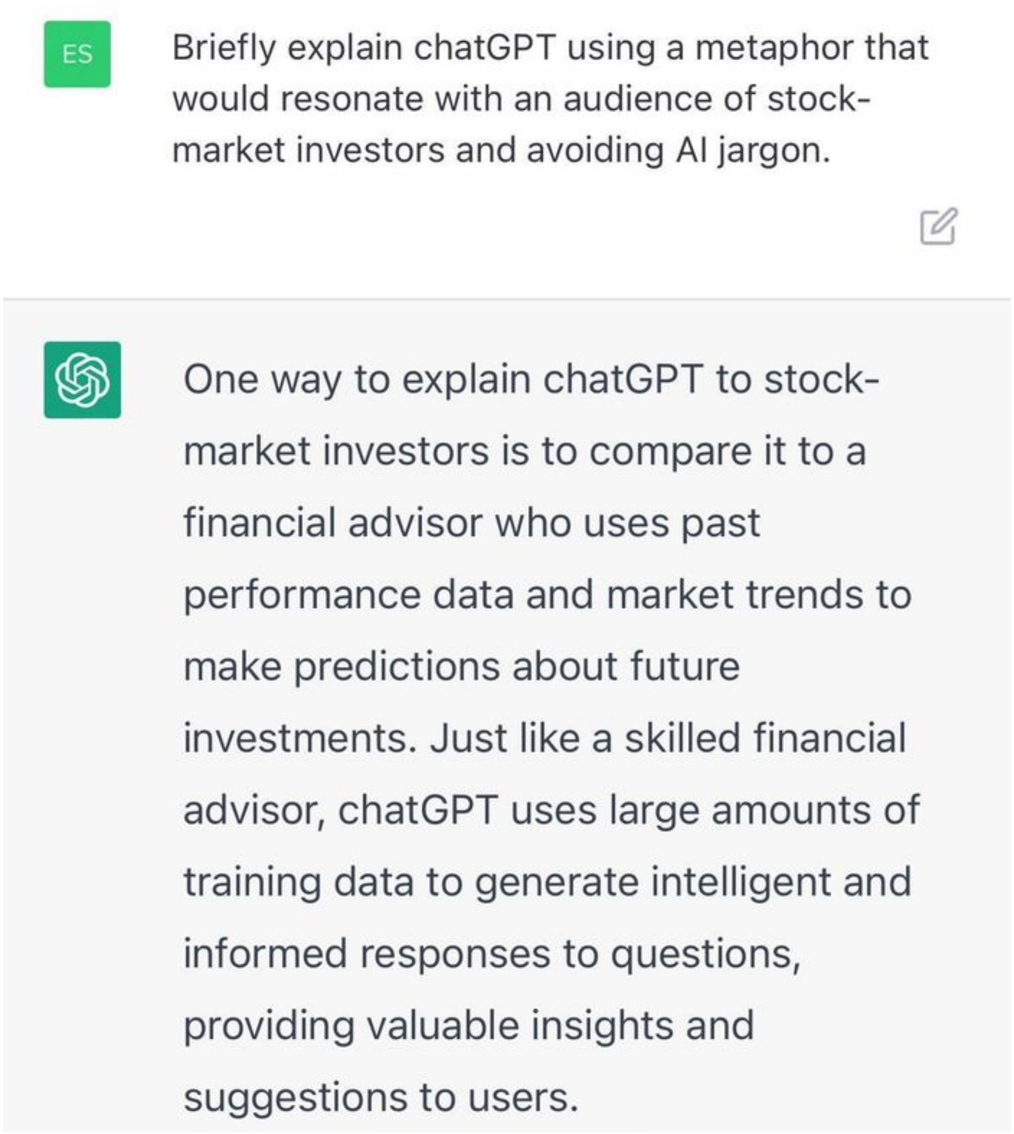
Tabular

Machine learning



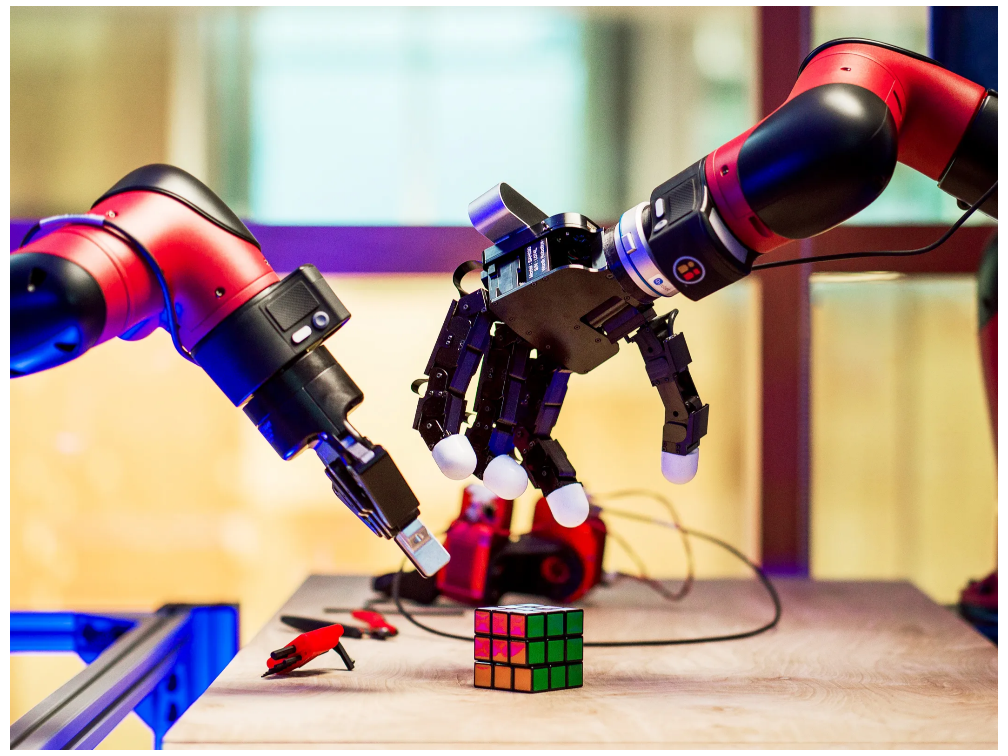
Visual

Computer vision



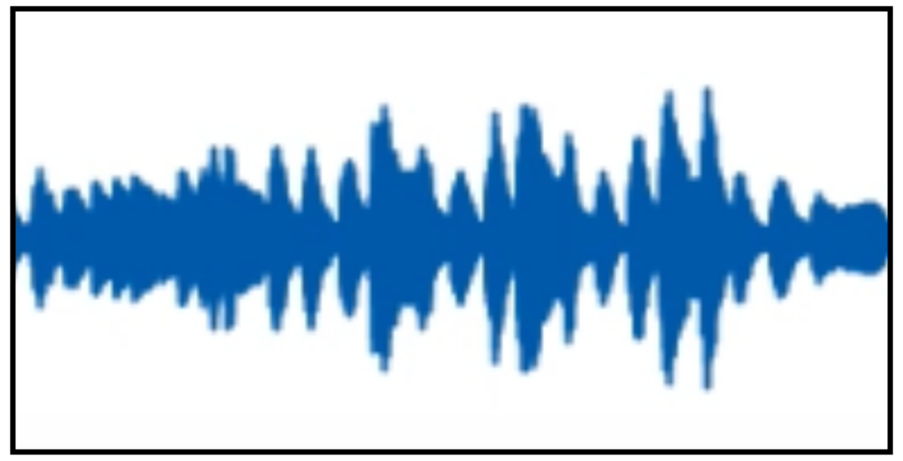
Text

Natural language processing (NLP)



Real manipulation

Robotics



Audio

Audio processing

What is machine learning?

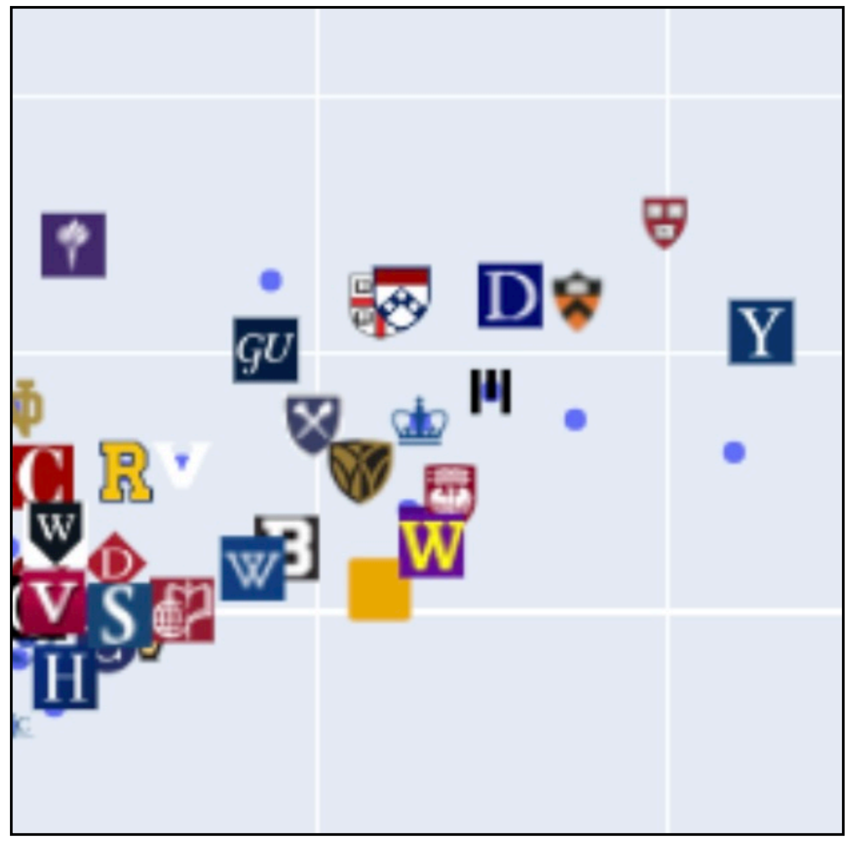
Machine learning (ML):
Algorithms that improve automatically using data or repeated experience.

Notes:

- This is **not** true for the programs that you have written so far in COS126 (i.e. your programs = explicit instructions for the computer).
- There are many non-ML algorithms in computer vision, NLP, robotics, etc.

Machine learning paradigms

1. Supervised learning: learn to **predict** an output from an input.
2. Unsupervised learning: learn **patterns** from data.
3. Reinforcement learning: learn by **interacting** with an environment (i.e. **trial-and-error**).



Supervised learning

Given (input, output) pairs, **predict** an output from its input.

$\text{input}_1 \rightarrow \text{output}_1$
$\text{input}_2 \rightarrow \text{output}_2$
$\text{input}_3 \rightarrow \text{output}_3$

Supervised learning

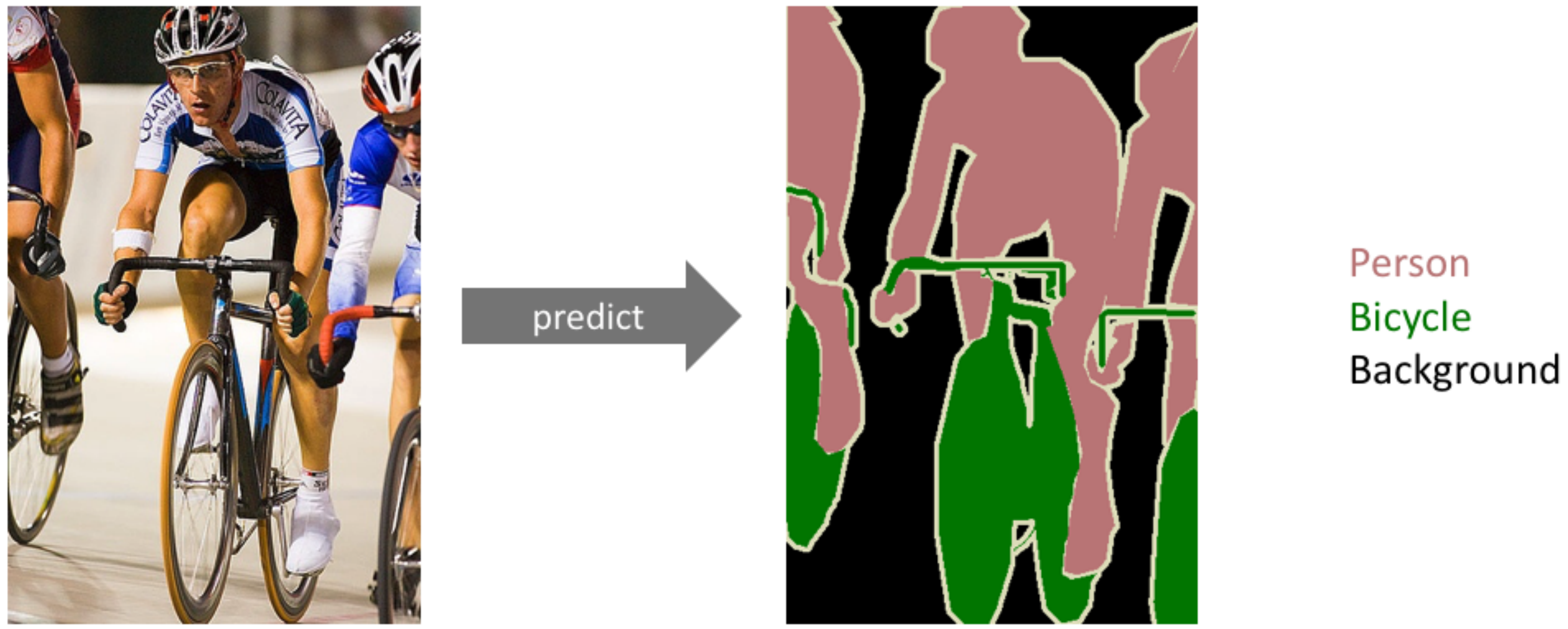
Medicine

- Input: medical information (i.e. age, body mass index [BMI], blood sugar levels, etc.)
- Output: blood pressure (in mm Hg)

Supervised learning

Computer vision: segmentation

- Input: an image
- Output: segmentation map of objects

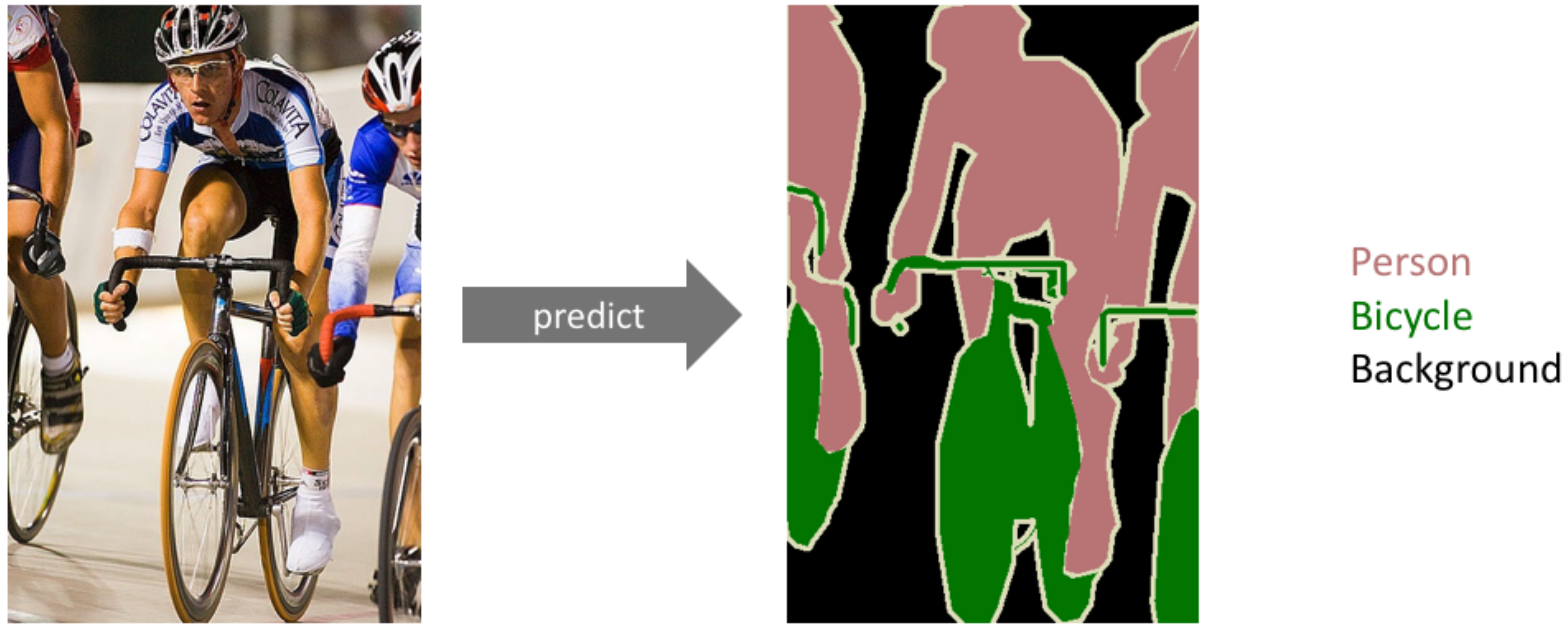


Pascal VOC dataset
http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

Supervised learning

Computer vision: segmentation

- Input: an image
- Output: segmentation map of objects
- **Application: self-driving cars**



Pascal VOC dataset
http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

https://github.com/meetps/pytorch-semseg

Supervised learning

NLP: Machine translation

- Input: a sentence in English
- Output: its translation in Spanish
- **Application: Google Translate**

Gender Bias in Machine Translation: https://towardsdatascience.com/gender-bias-in-machine-translation-819ddce2c452

"That river is dangerous to swim in." → "Es peligroso nadar en ese río."

"The baby is playing with some toys." → "El bebé juega con algunos juguetes."

Anki Tab-Delimited Bilingual Sentence Pairs
https://www.manythings.org/anki/

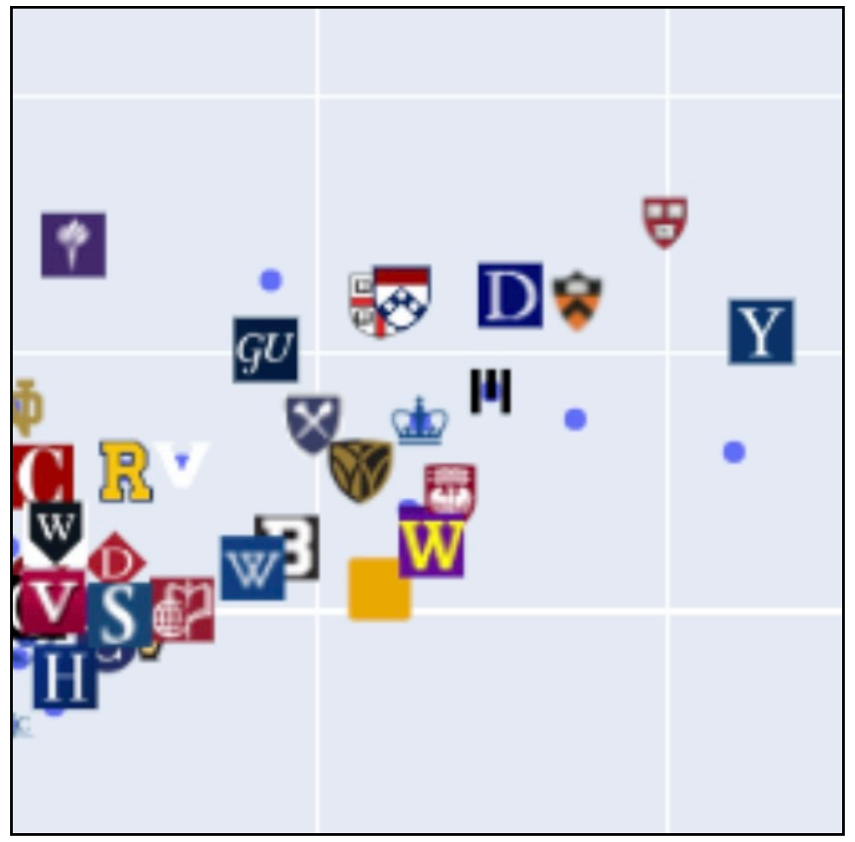
Supervised learning

Other applications

- Spam filtering: email text → {spam, not spam}
- Speech recognition: audio clip → text transcription
- Medical imaging: CT scan of COVID patients → severity of COVID symptoms
- …

If you can create a dataset of (input, output) pairs, you can use supervised learning.

Unsupervised learning



Learn **patterns** from unlabelled data (i.e. no input-output pairs).

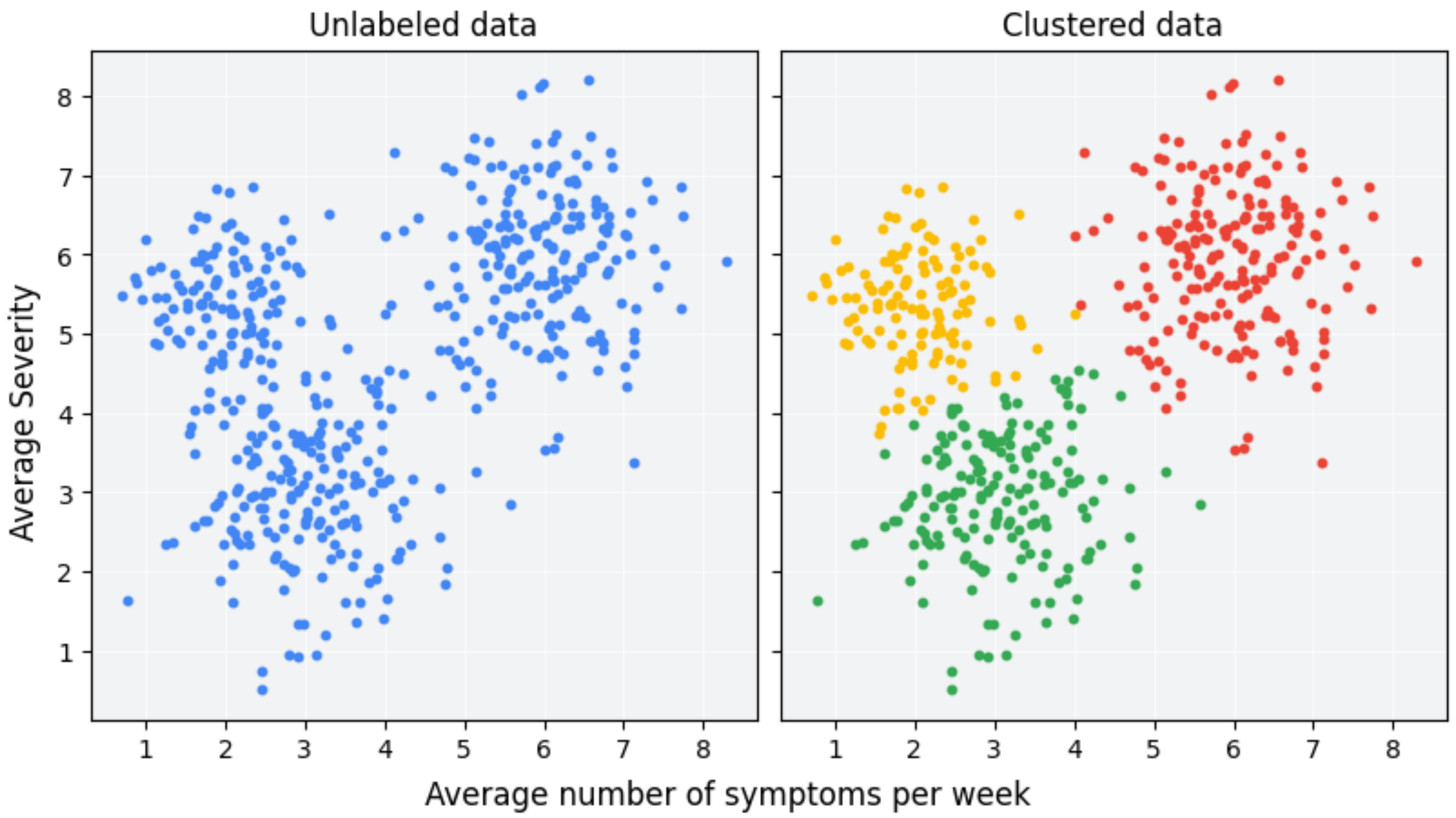
Examples:

- Clustering
- Dimensionality reduction
- Anomaly detection
- Generative modeling
- …

Unsupervised learning

Clustering

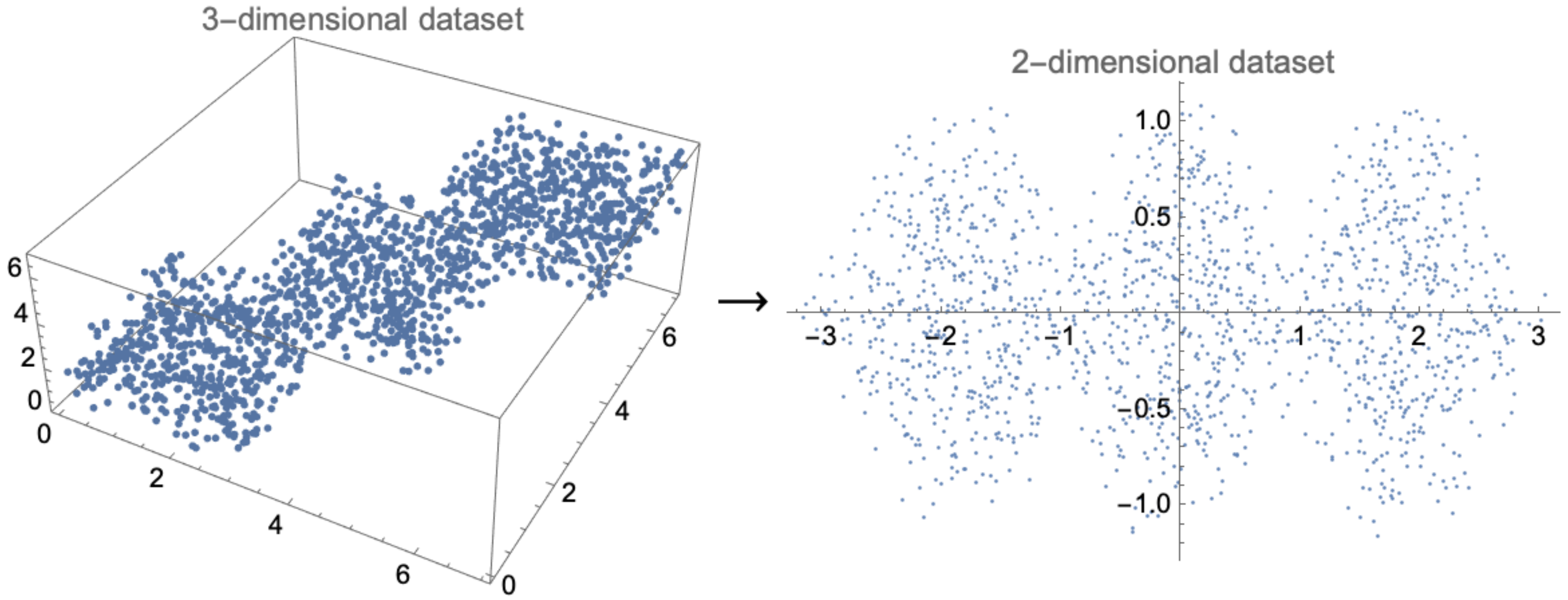
- Goal: Given a data point, categorize it into one of k clusters.
- Unlike supervised learning, there are no output labels! i.e. no (data point, cluster) pairs
- Application in medicine: What types of cancers are similar to each other?



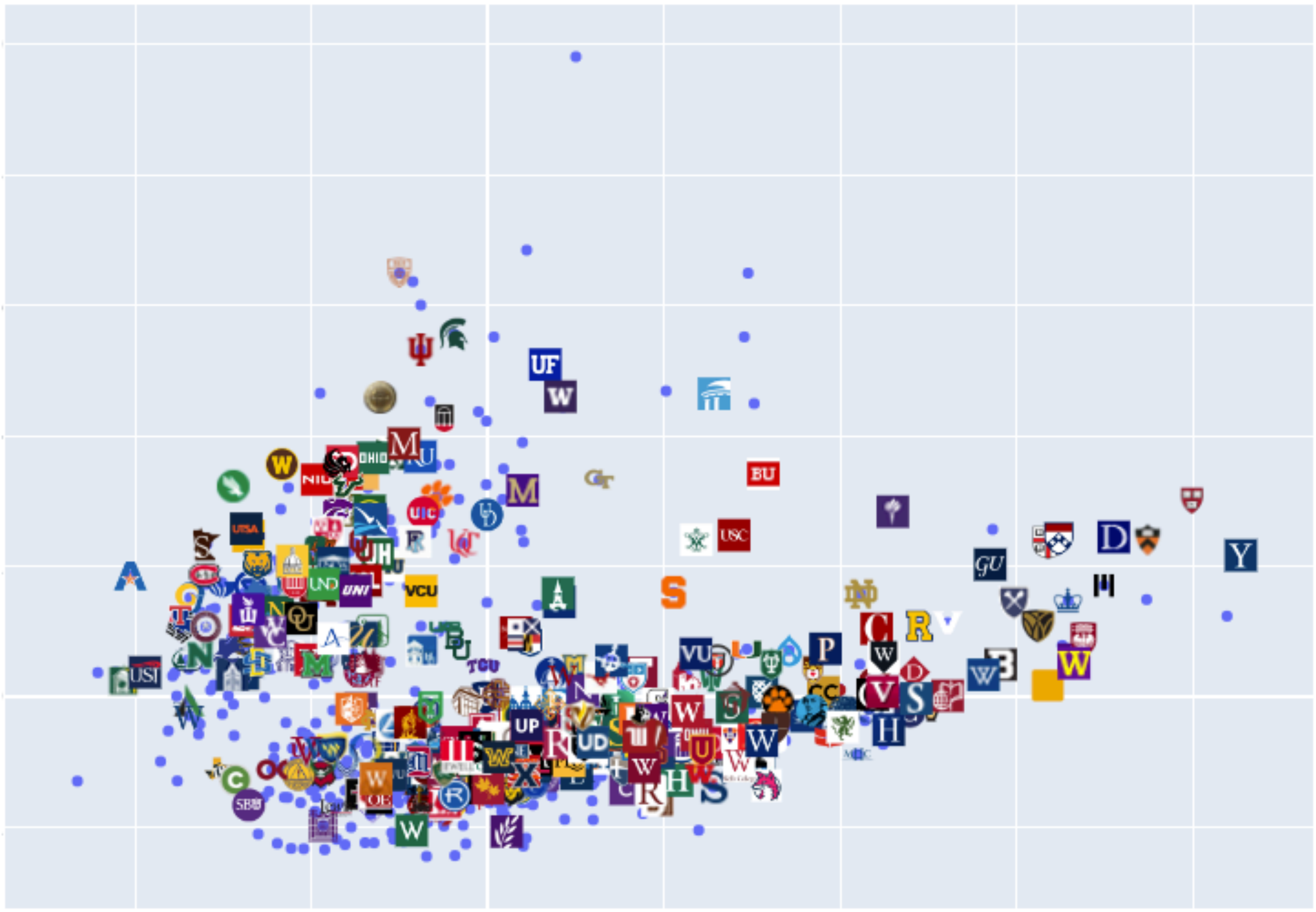
Unsupervised learning

Dimensionality reduction

- Goal: Reduce number of variables while preserving patterns (e.g. distances between points)
- Uses:
  - Compression
  - Visualization
  - …



PCA example of colleges dataset

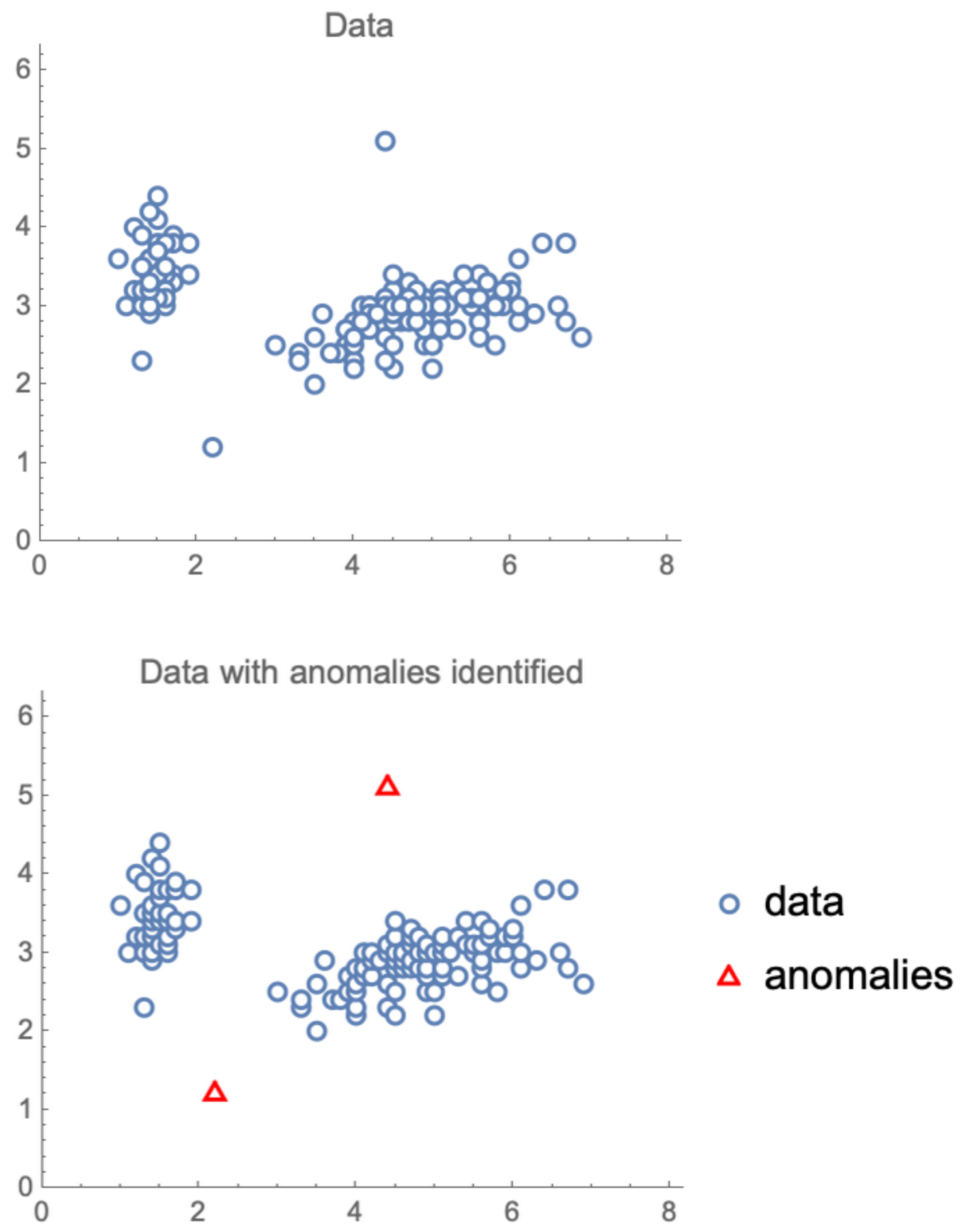


https://www.ruthfong.com/cos324/pca_colleges.html

Unsupervised learning

Anomaly detection

- Goal: Identify anomalies (outliers)
- Applications:
  - Credit card fraud
  - Medical diagnosis
  - …



Unsupervised learning

Generative model

- Goal: Create synthetic examples that look realistic.
- Applications:
  - Computer vision:
    Given a dataset of portraits, generate realistic looking faces.
  - NLP:
    Given Shakespeare's sonnets, generate a realistic sounding, Shakespearean sonnet.

Generated by StyleGAN
(Karras, et al., CVPR 2019)

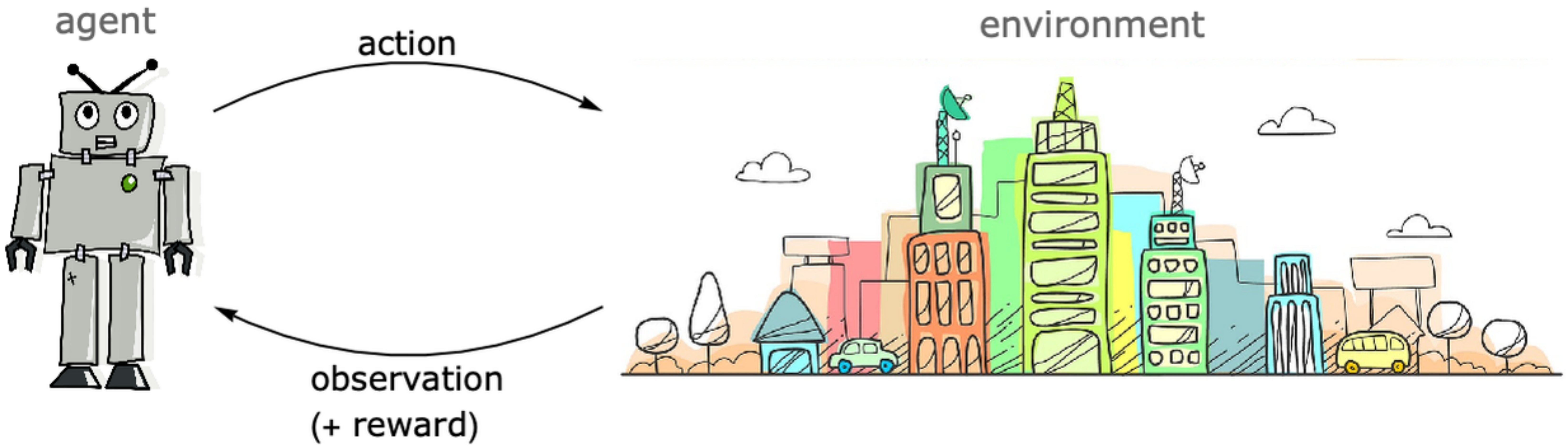
*"am of my faults thy sweet self dost deceive:
they look into the beauty of thy deeds
nativity, once in the chronicle of wasted time
not from the thing they see"*

Generated by Markov Chains
https://rpubs.com/malcolmbarrett/shakespeare

Reinforcement learning

Learn by **interacting** (i.e. **trial-and-error**) with an *environment.*

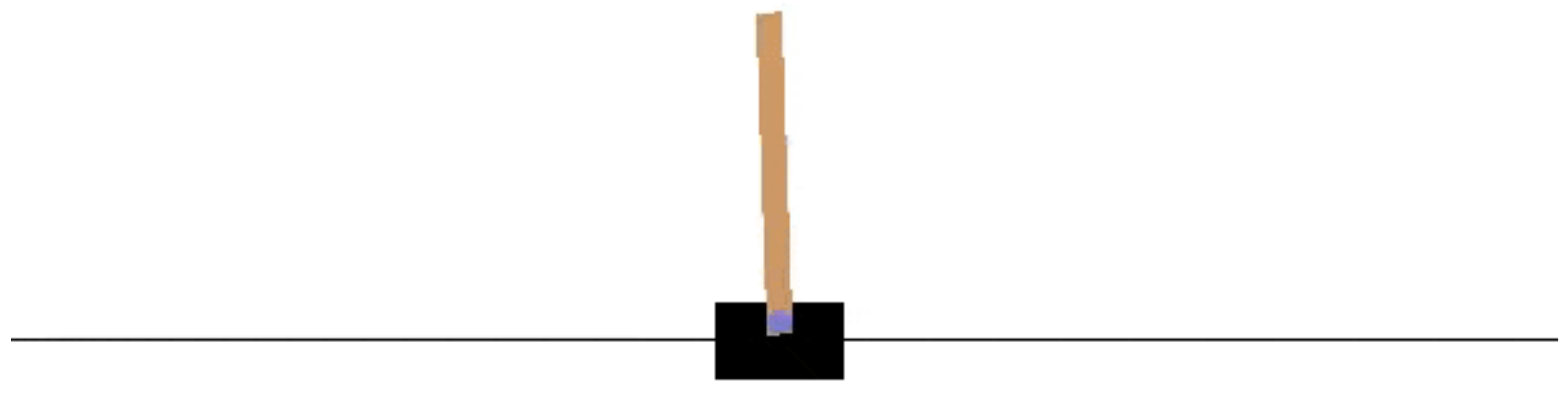
- An *agent* interacts with an *environment* by performing *actions* and then gathers *observations* (and sometimes *rewards*) that occur in response to its actions
- Interactions generate data (i.e. no fixed dataset).
- Goal: Learn *agent* to maximize *reward*.



Reinforcement learning

CartPole

- Goal: Balance a pole on a cart by moving it left and right.
- Actions: {left, right}
- Reward: +1, for each step the pole is upright.
- Learn over many attempts (i.e. episodes).

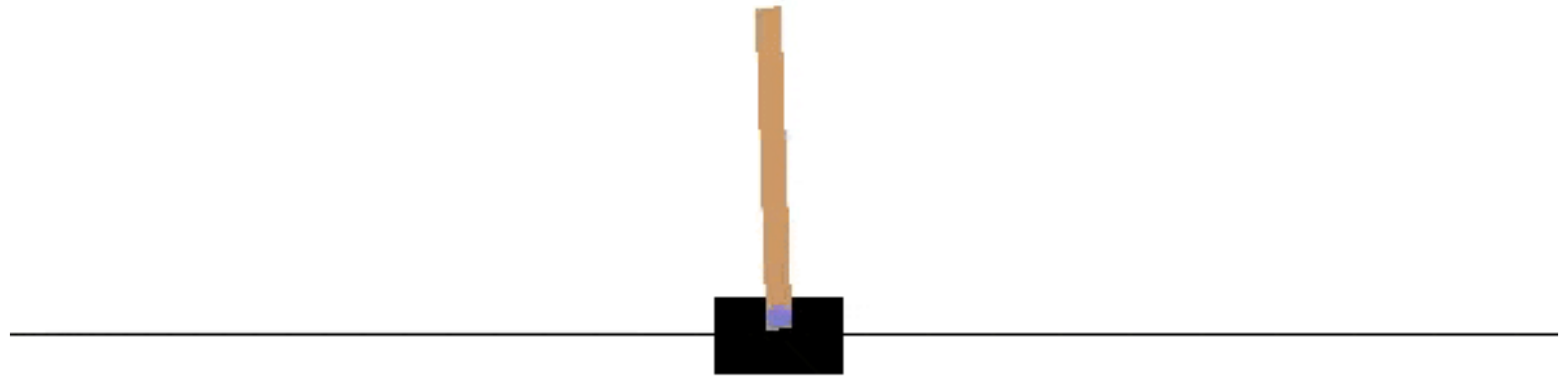


https://gymnasium.farama.org/environments/classic_control/cart_pole/

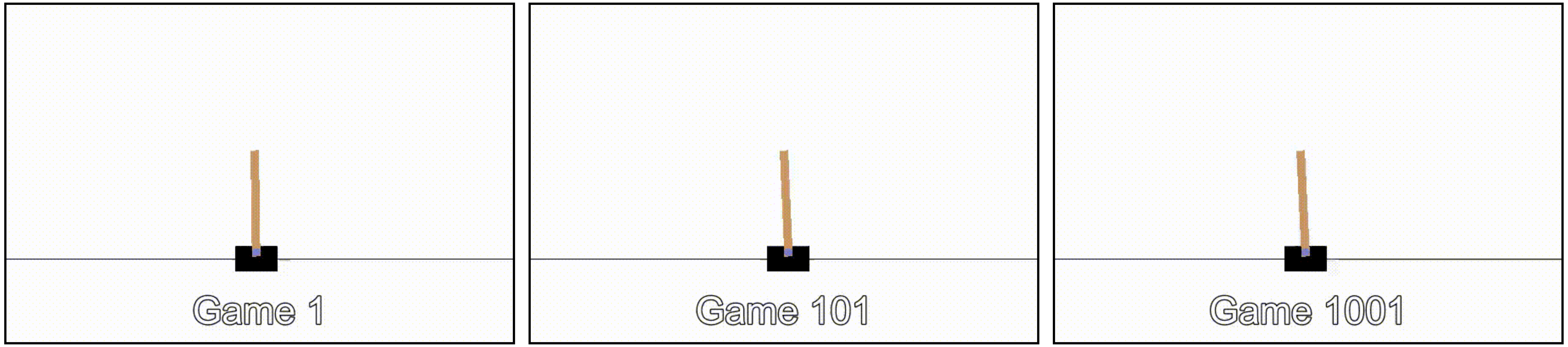
Reinforcement learning

CartPole

- **Goal:** Balance a pole on a cart by moving it left and right.
- **Actions:** {left, right}
- **Reward:** +1, for each step the pole is upright.
- Learn over many attempts (i.e. episodes).



https://gymnasium.farama.org/environments/classic_control/cart_pole/

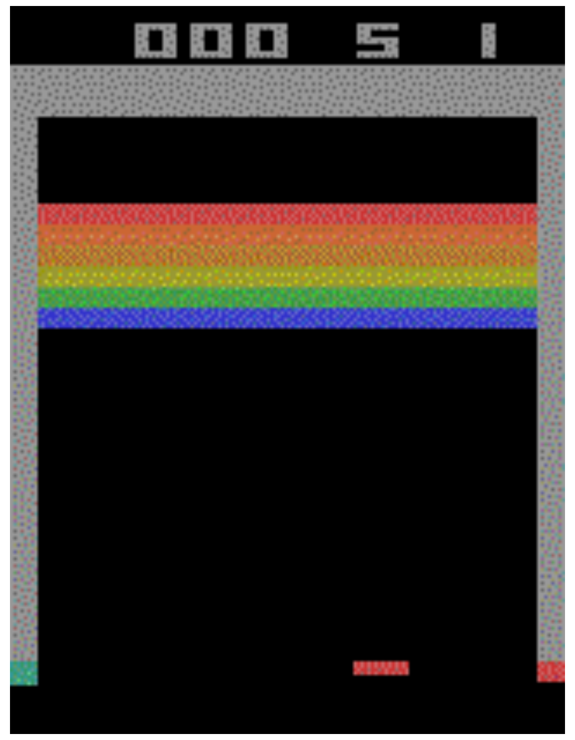


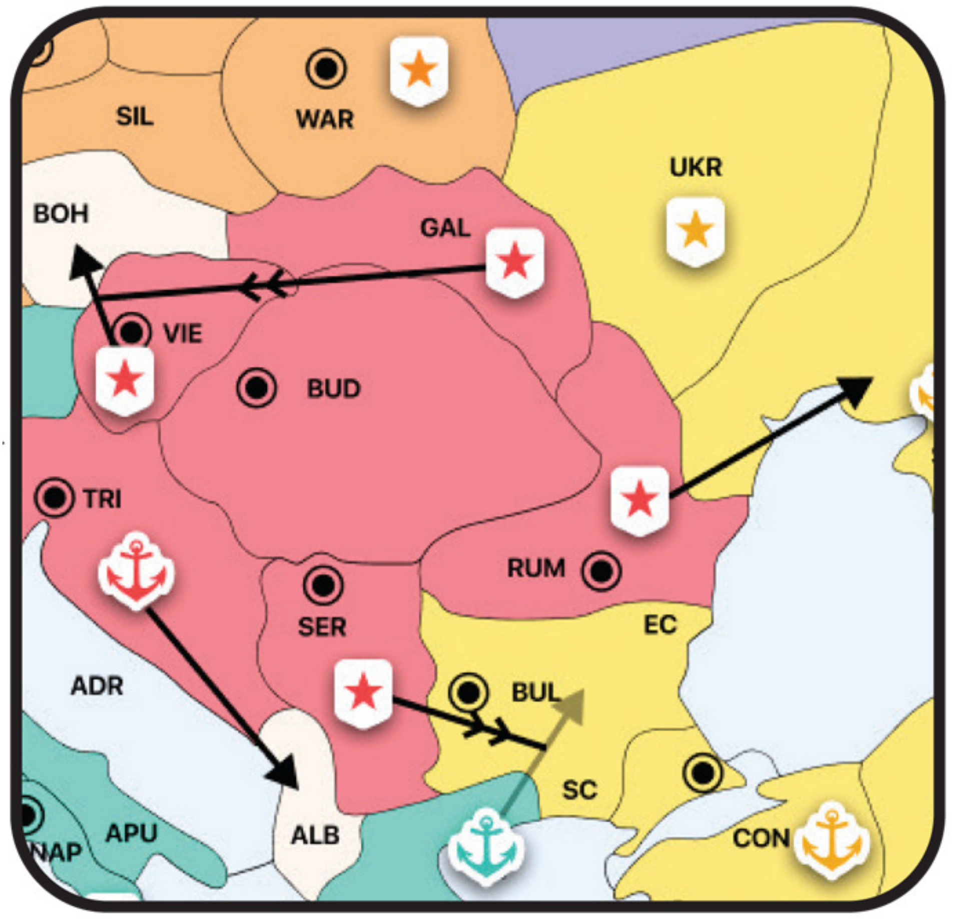
https://blog.ml6.eu/importance-of-data-visualization-for-ml-936239c355d0

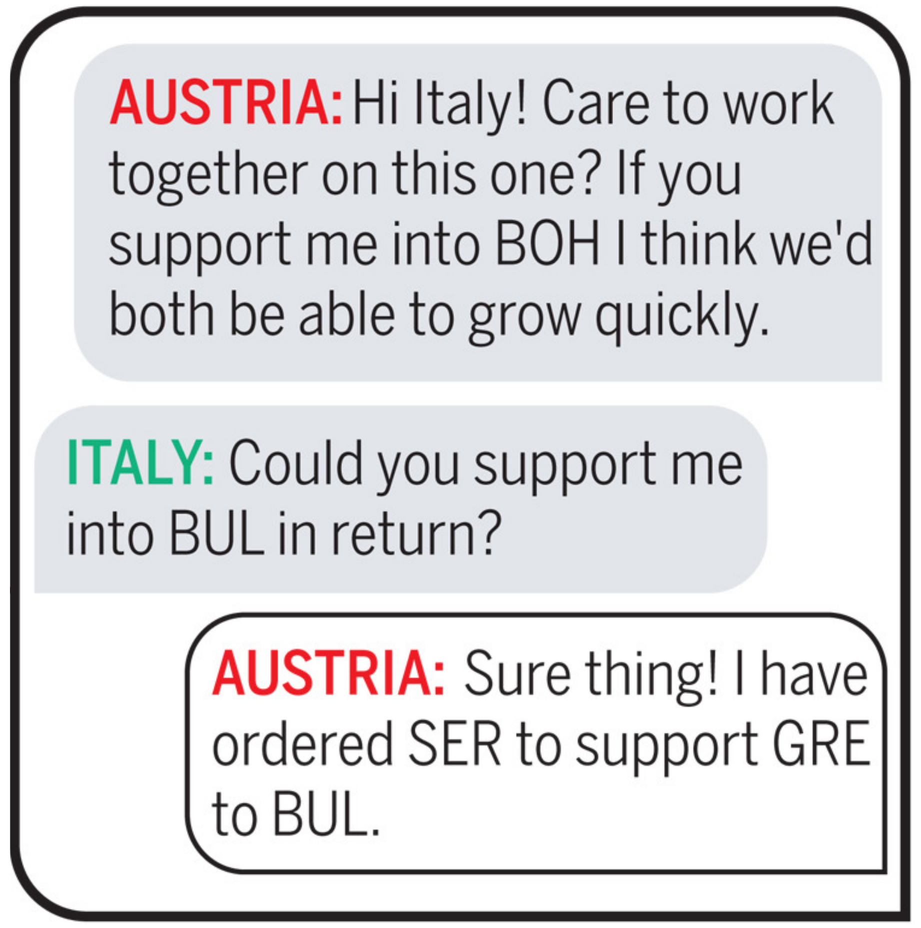
Reinforcement learning

Application: Games

- Atari (2015): DeepMind
- Go (2016): AlphaGo from DeepMind
- Poker (2017, 2019):
  Libratus and Pluribus from CMU
- Diplomacy (2022): Cicero from Meta AI



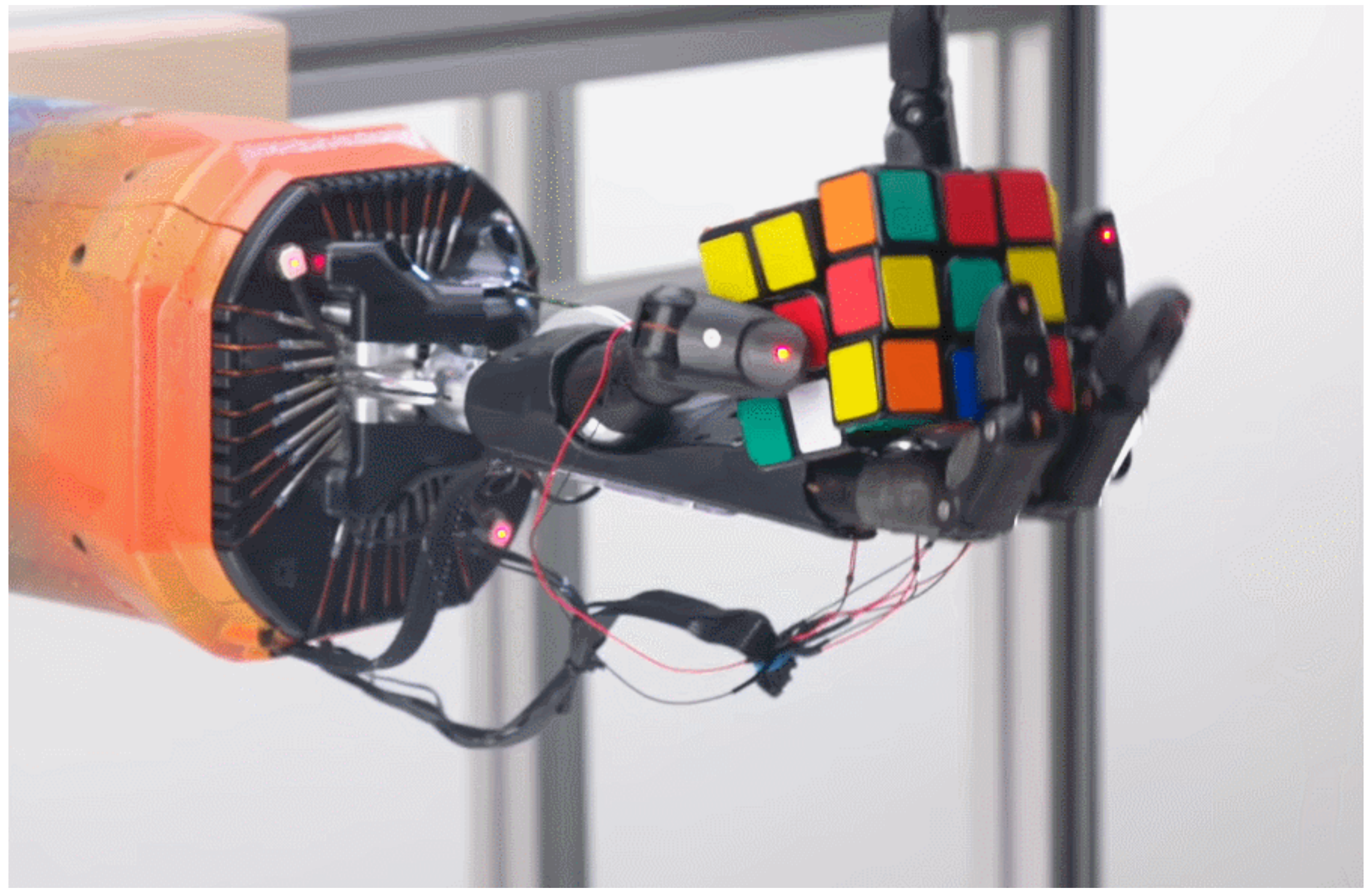




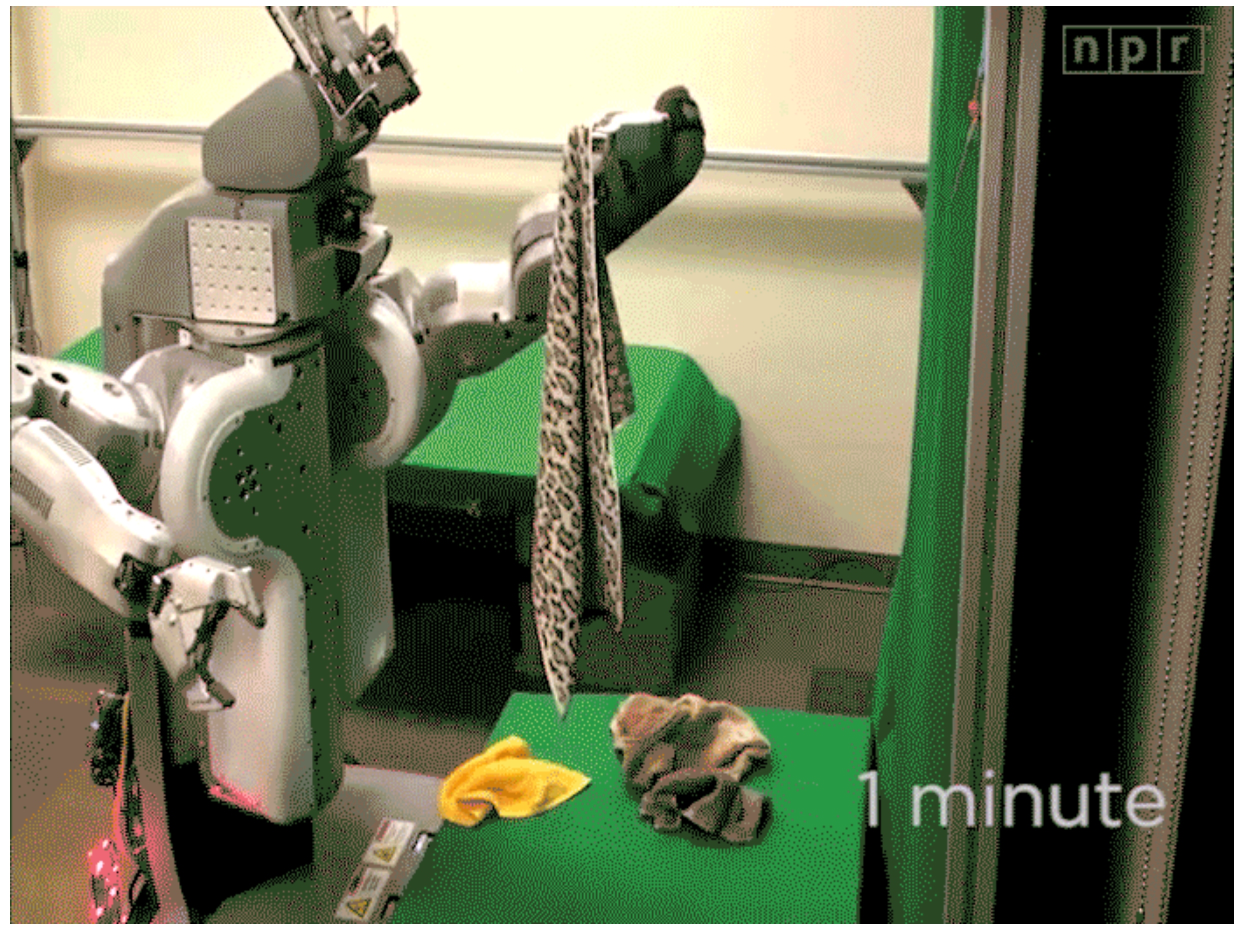
Reinforcement learning

Application: Robotics

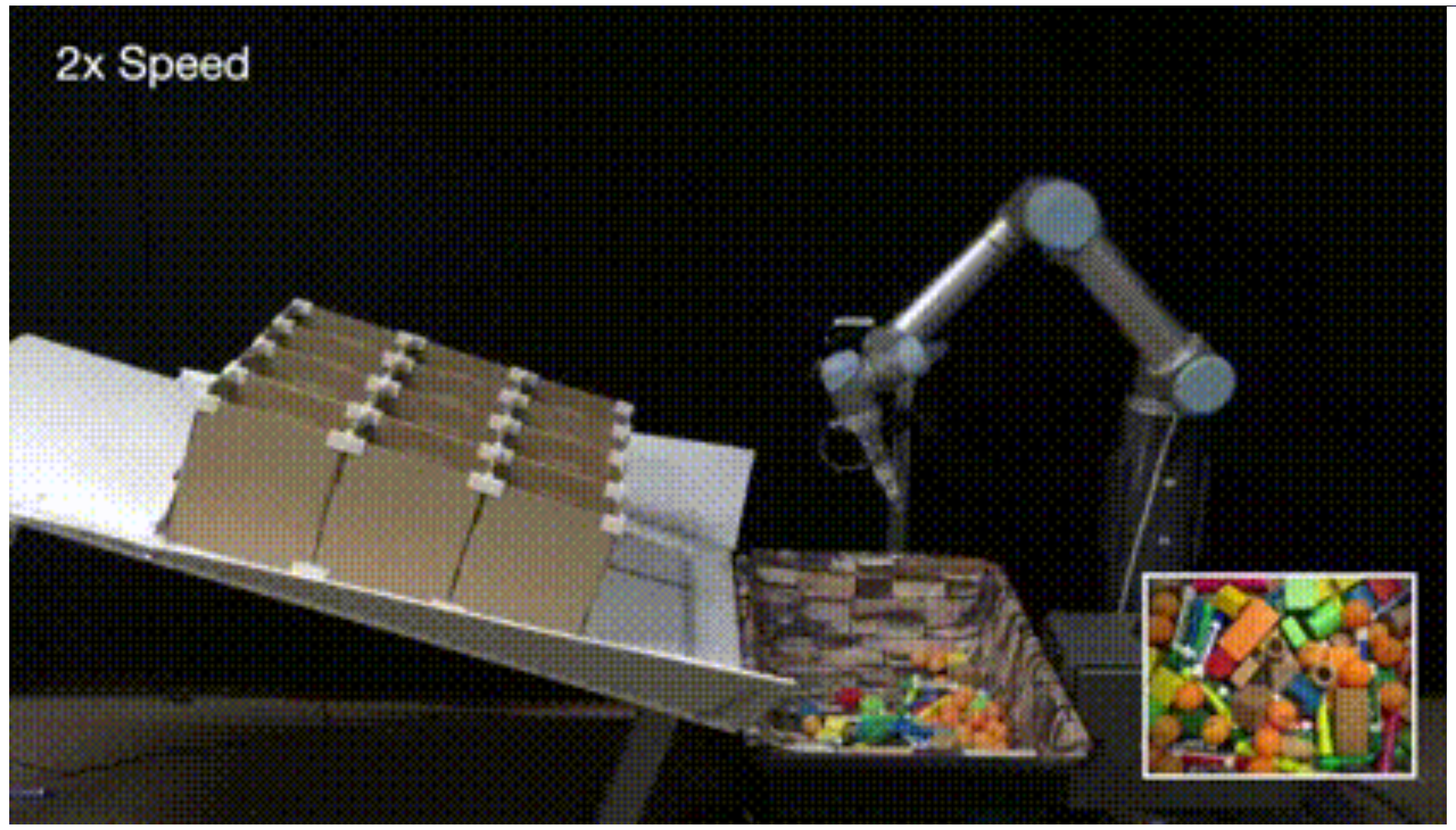
- Solving a Rubik's cube
- Manipulating objects
  (e.g. grasp, throw, etc.)
- Laundry folding



OpenAI's ADR (2019)



Berkeley's BRETT (2015)



Princeton+Google's TossingBot (2019)

Poll

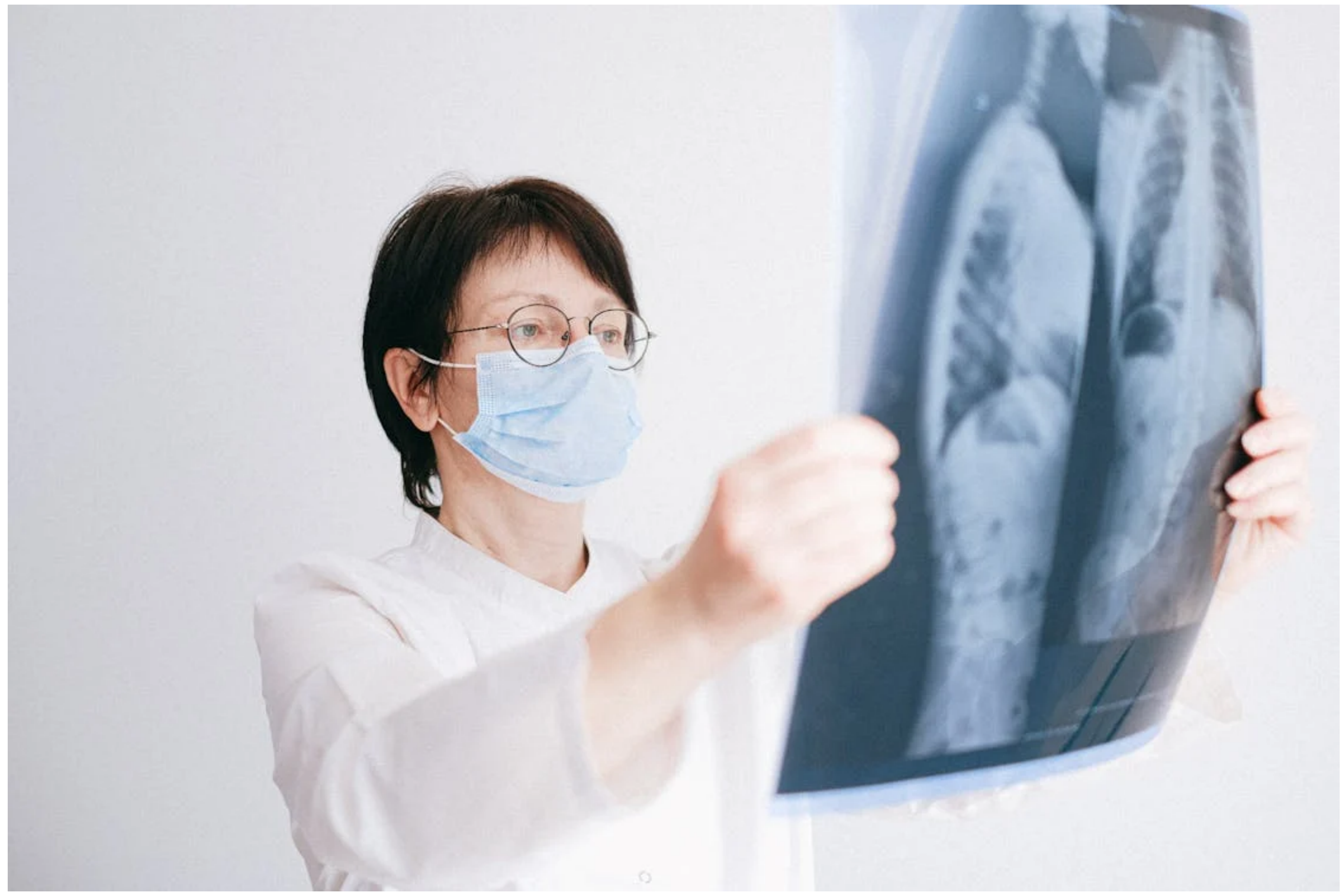
Predict if a patient's lung scan is abnormal based on a huge dataset of lung scans.

What kind of machine learning problem is this?

A. Supervised learning

B. Unsupervised learning

C. Reinforcement learning



https://www.pexels.com/photo/doctor-looking-at-lung-scans-she-is-holding-4225926/

Machine learning paradigms

1. Supervised learning: learn to **predict** an output from an input.
2. Unsupervised learning: learn **patterns** from data.
3. Reinforcement learning: learn by **interacting** with an environment (i.e. **trial-and-error**).

Other paradigms:

- *Self-supervised learning*
- *Semi-supervised learning*
- *Transfer learning*
- *Online learning*
- *…*

Ingredients of ML

1. **Data.** Can be labelled, unlabelled, or from experience.
2. **Model.** A model maps from datapoint to a desired answer or output.
3. **Model parameters**. Internal parameters of model that are learnable.
4. **Training.** Given datapoints, find good model parameters.
5. **Testing.** Evaluate the performance of learned model
   ... on new, previously unseen data (i.e. test set).

Perceptron

Supervised learning

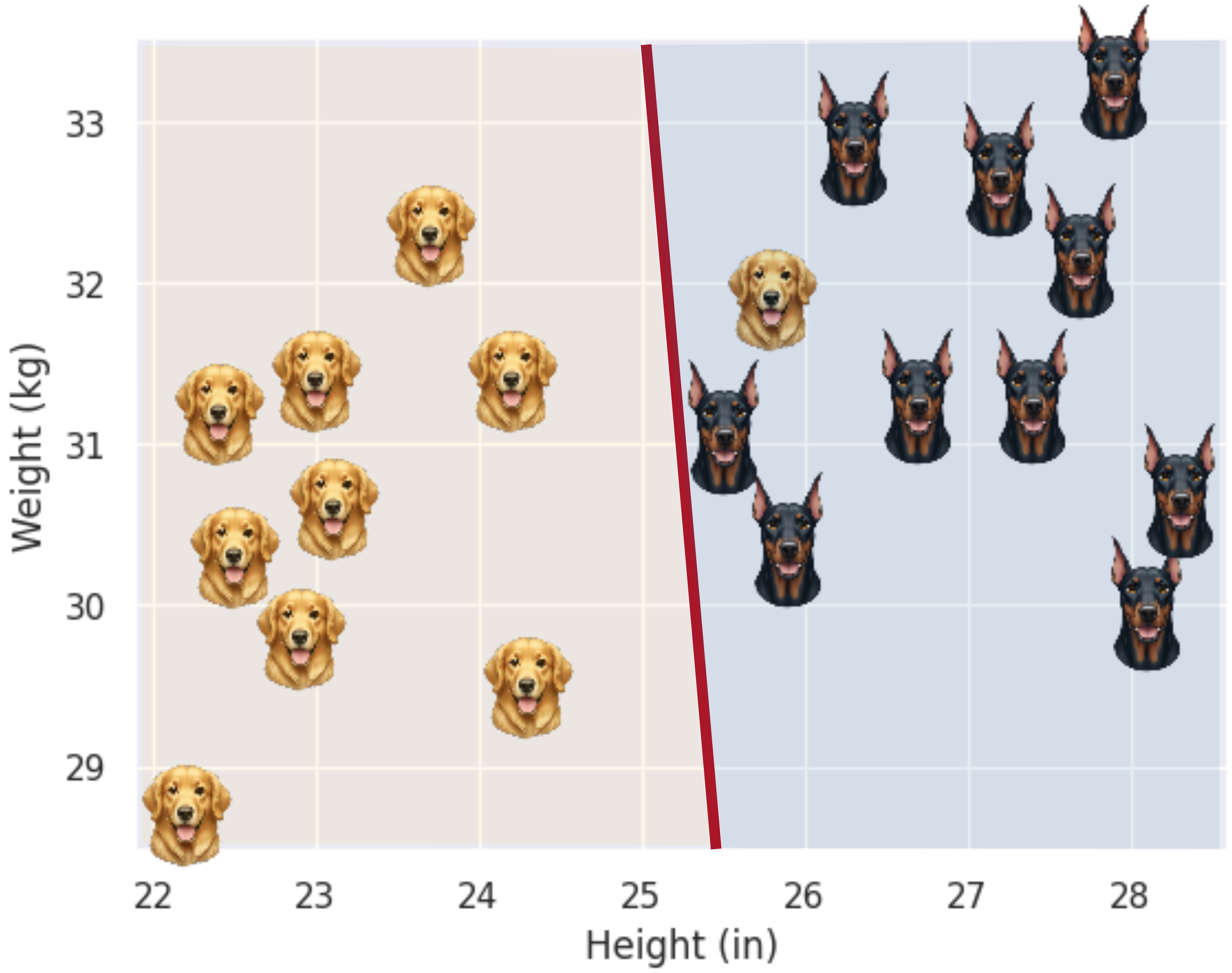
Given (input, output) pairs, **predict** an output from its input.

$\text{input}_1 \rightarrow \text{output}_1$
$\text{input}_2 \rightarrow \text{output}_2$
$\text{input}_3 \rightarrow \text{output}_3$

Basic problems described by output type:

- **Regression**: Predict a continuous value (e.g. number).
- **Classification**: Predict a discrete category (i.e. classes).
  - **Binary classification**: Predict one of **two** classes.
  - **Multi-class classification**: Predict one of **k** classes (k > 2).

Binary classification



Data is nearly **linearly separable**.

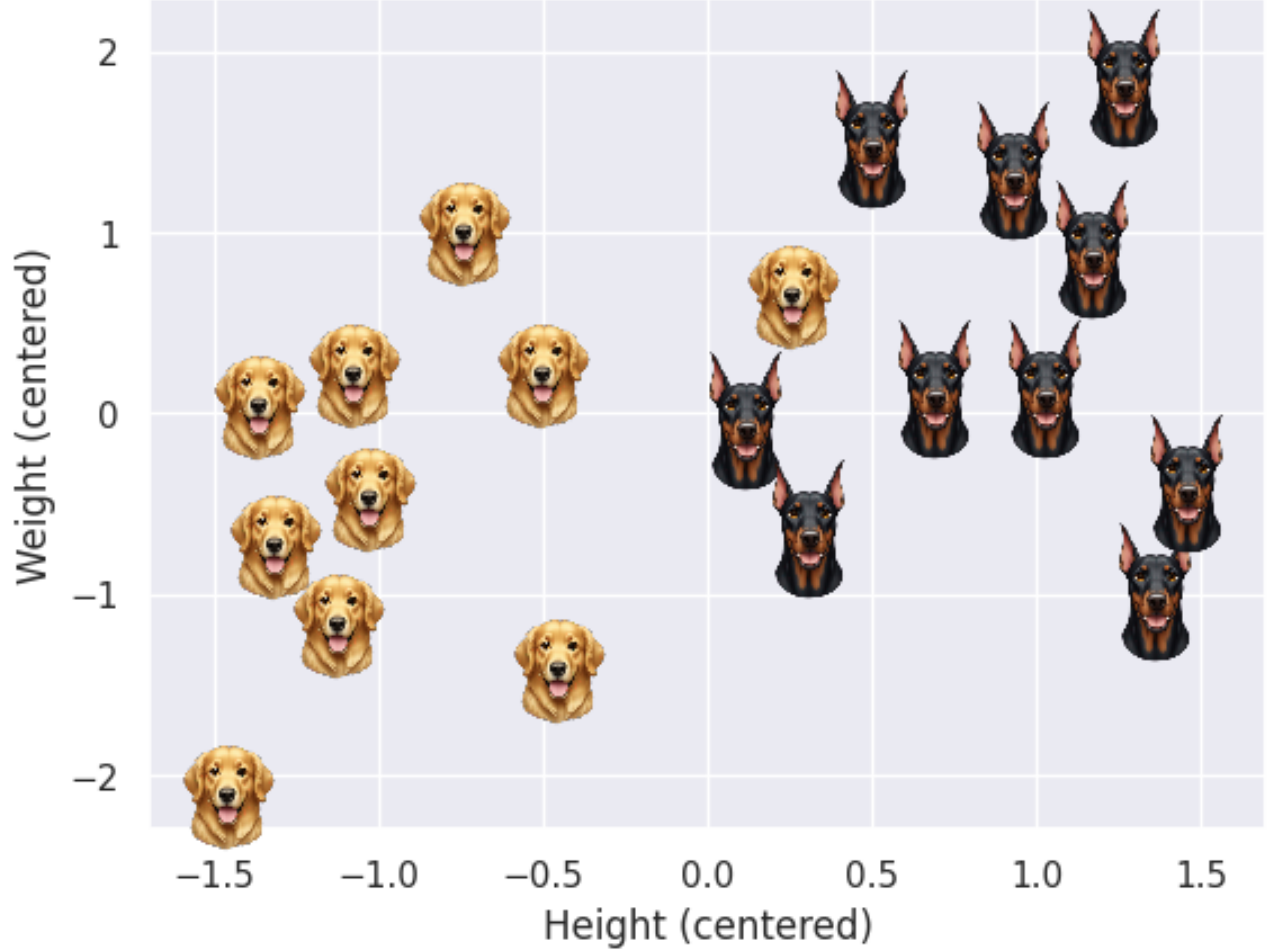
Binary classification

Predict dog species

- Input (centered*)
  - Height
  - Weight
- Output
  - Golden Retriever: $+1$
  - Doberman Pinscher: -1

= "Golden Retriever detector"

* Input centered to have
mean = 0 and standard deviation = 1.

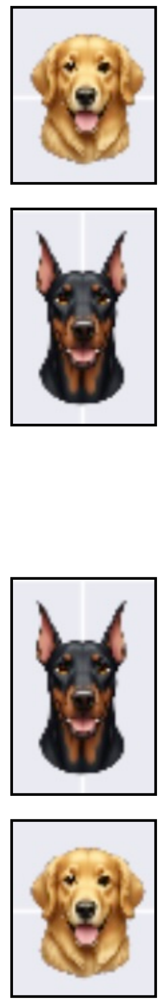


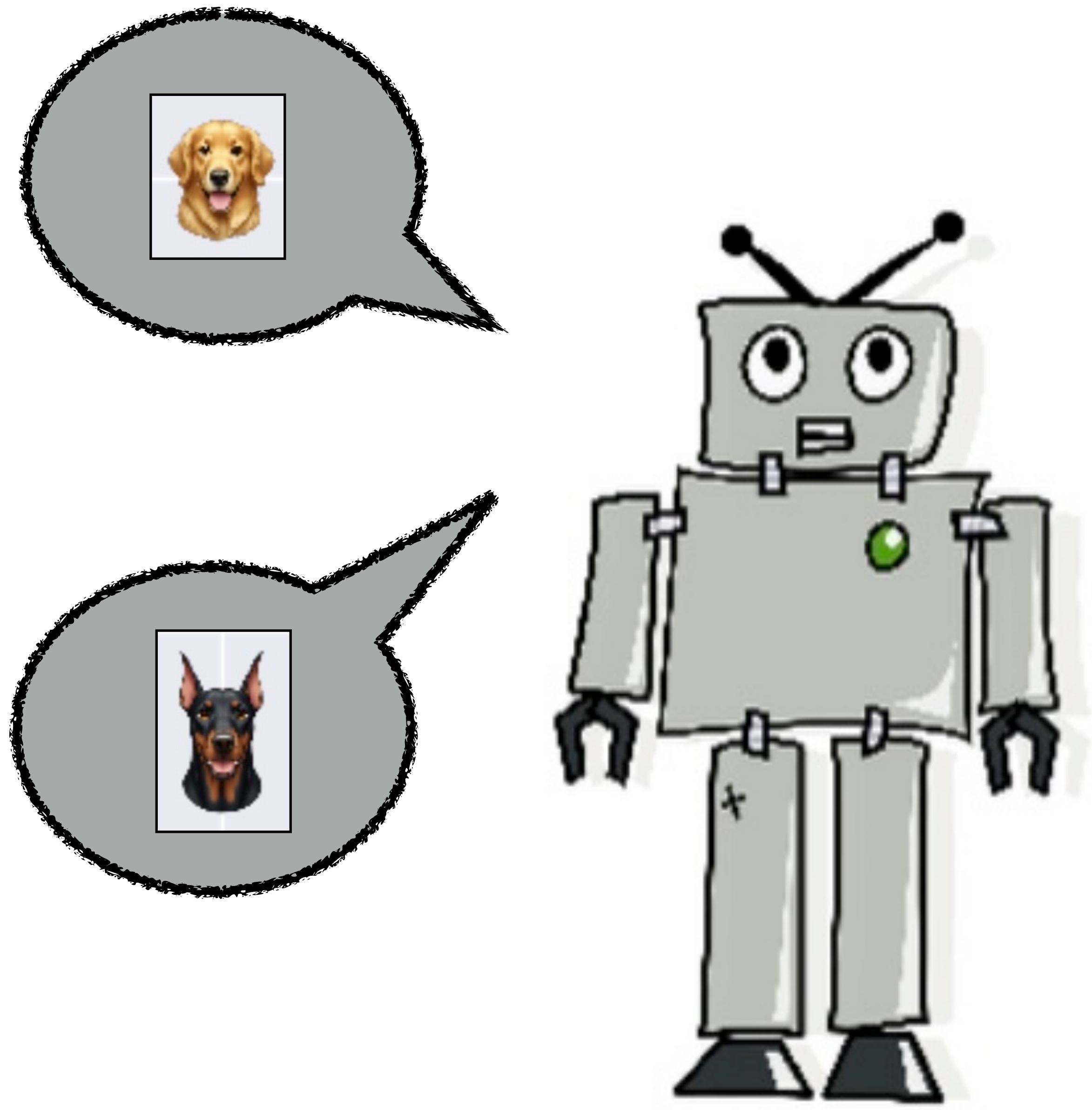
True/False Positive/Negative

Ways to get the answer right or wrong:

- True Positive
- False Positive

- True Negative
- False Negative

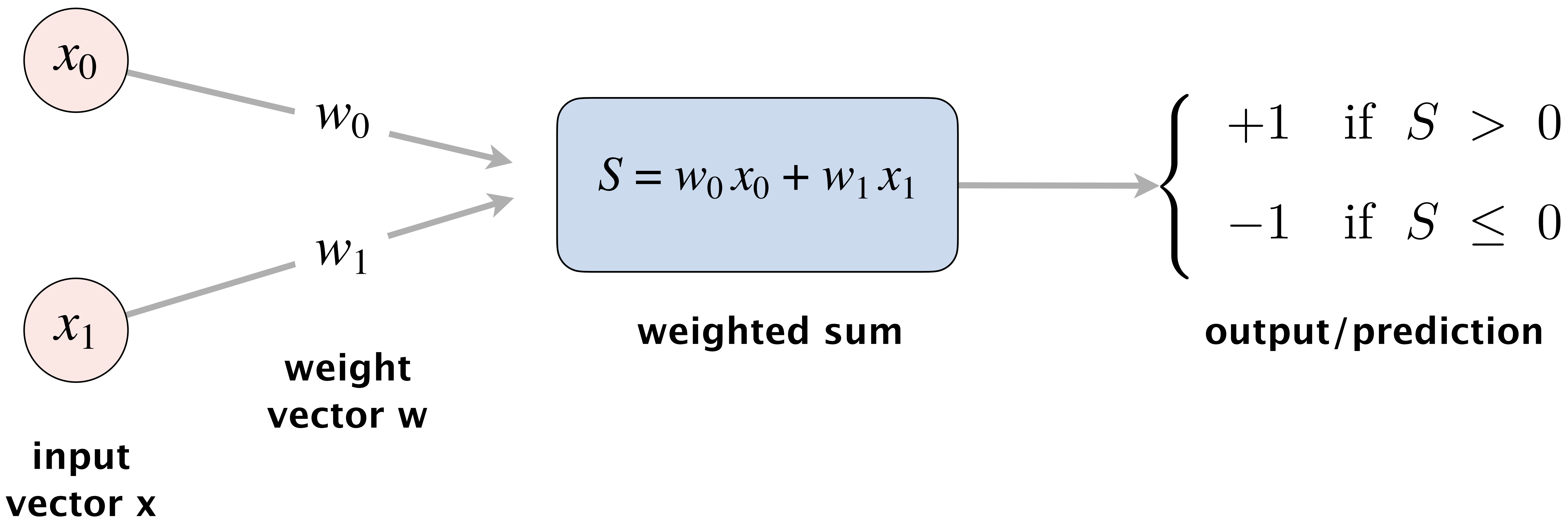




Perceptron

Algorithm that takes several inputs and produces a single, binary output.

- Inputs: x_0, x_1 (i.e. height, weight)
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1$
- Output
  $\text{sign}(S) = +1$ if $S > 0$
  $= -1$ if $S \leq 0$



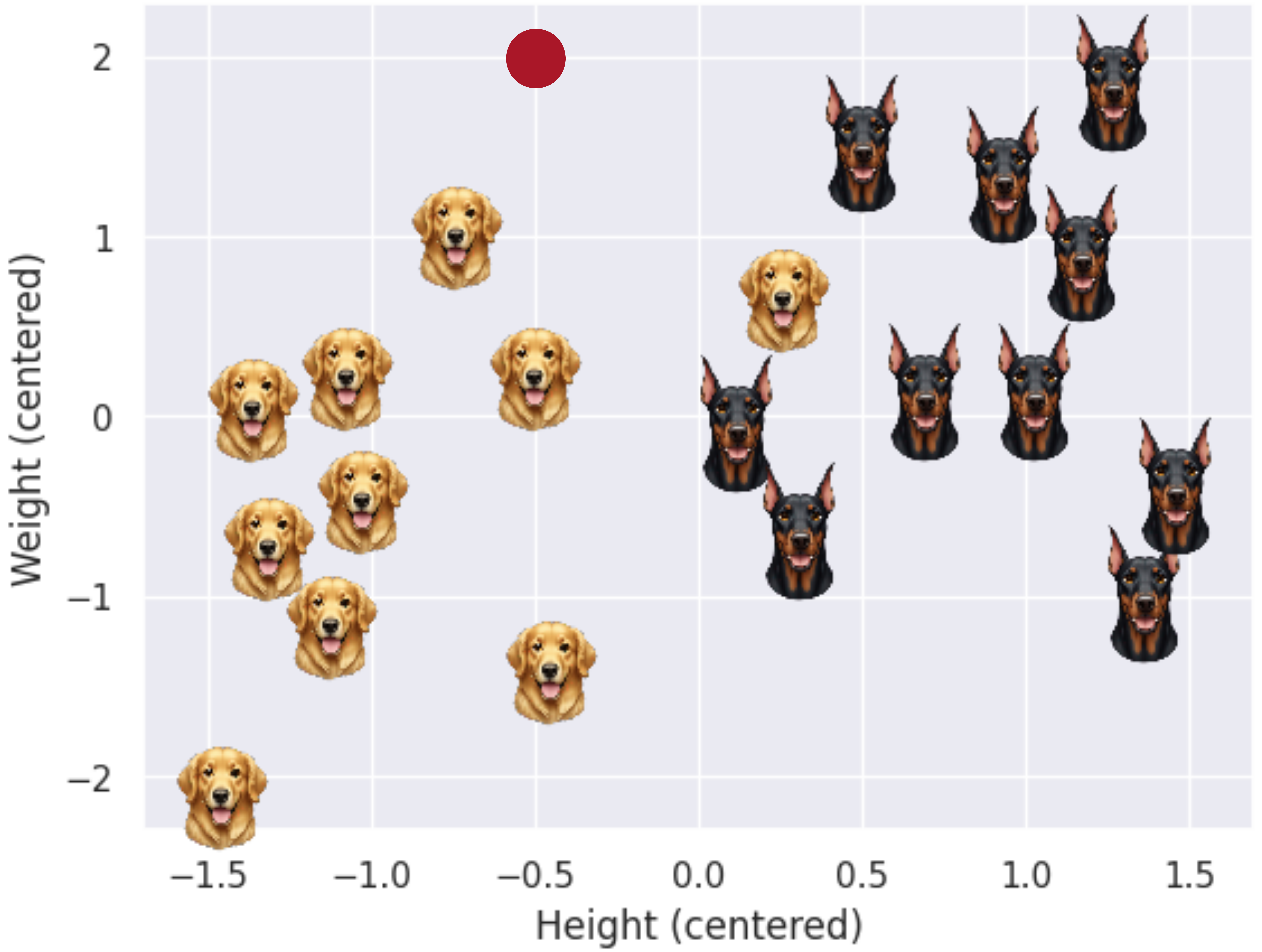
Poll

- Inputs: $x_0 = -0.5,\ x_1 = 2$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1$
- Output
  $\text{sign}(S) = +1$ if $S > 0$
  $= -1$ if $S \leq 0$

What is the weighted sum S?

A. -1.2 C. 0.8

B. -0.2 D. 1.0



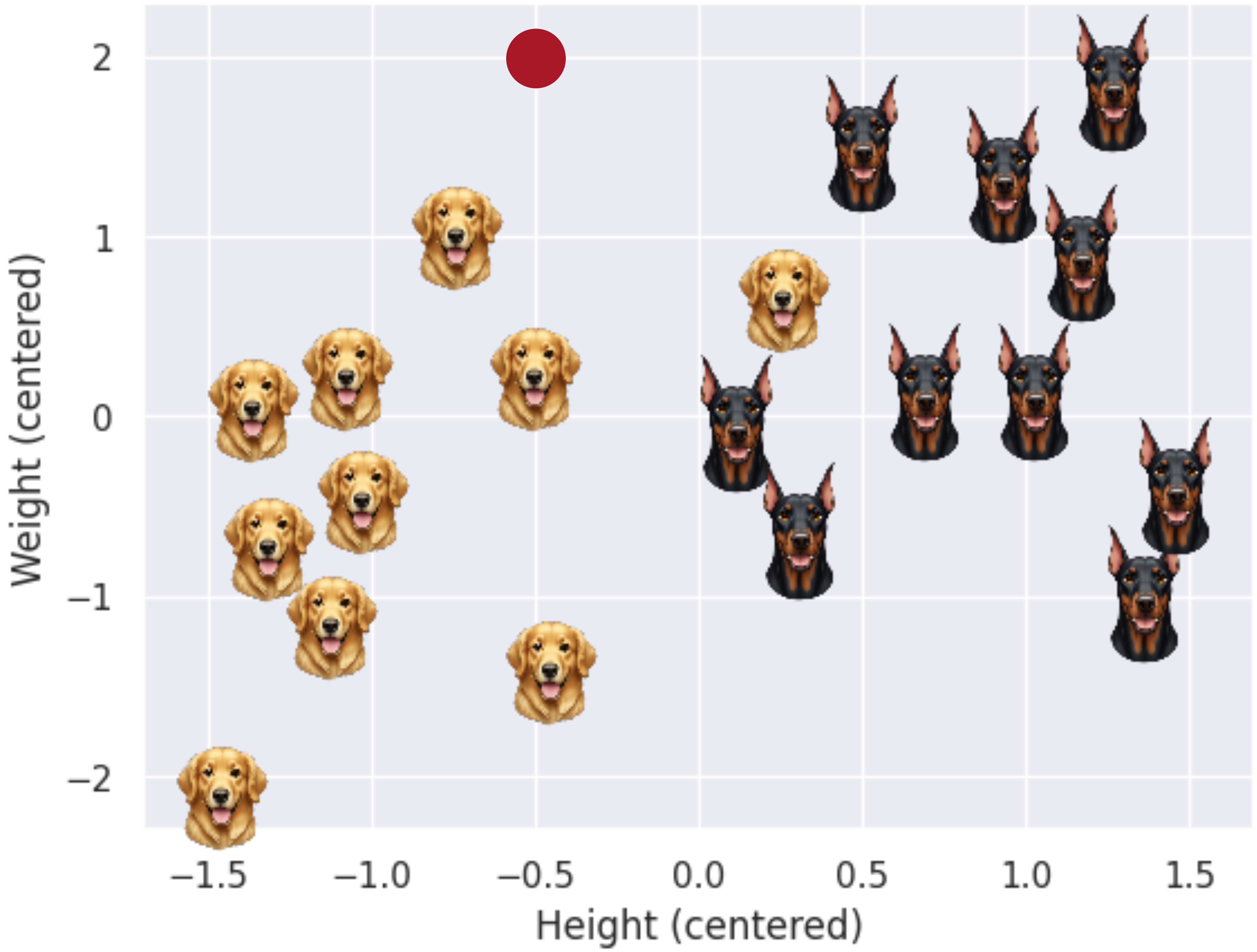
Golden retriever: +1, Doberman -1

Perceptron

- Inputs: $x_0 = -0.5,\ x_1 = 2$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1$
- Output
  $\text{sign}(S) = +1$ if $S > 0$
  $\qquad\qquad = -1$ if $S \leq 0$

$$S = w_0 x_0 + w_1 x_1$$
$$= (-2) \cdot (-0.5) + (-0.1) \cdot 2$$
$$= 1 + (-0.2) = 0.8$$

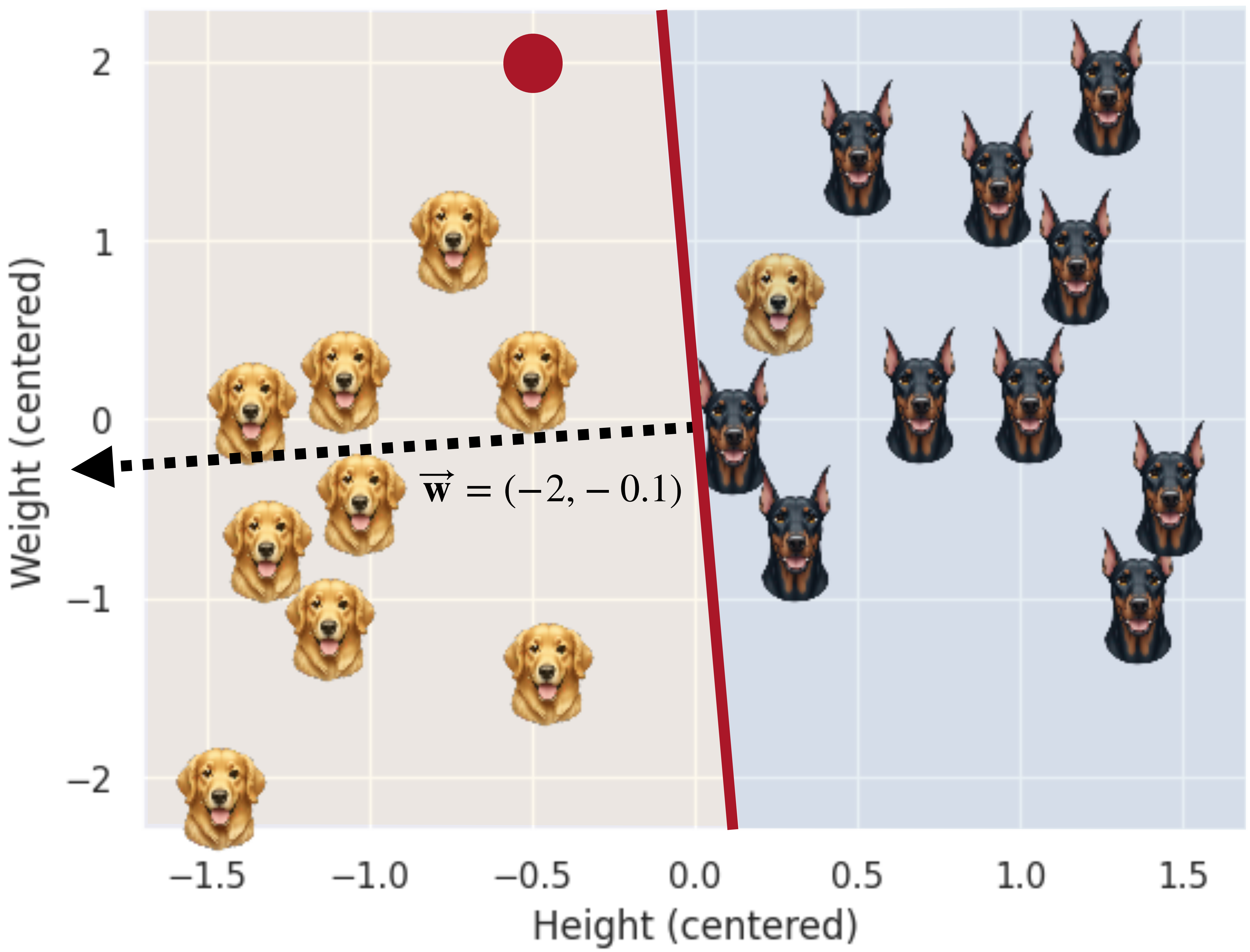
$\text{sign}(S) = \text{sign}(0.8) = +1$



Golden retriever: +1, Doberman -1

Perceptron: Geometric Perspective

- Inputs: $x_0 = -0.5,\ x_1 = 2$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute weighted sum
  $S = w_0 x_0 + w_1 x_1 = 0.8$
- Output
  $\text{sign}(S) = +1$
- Weight vector $\vec{\mathbf{w}}$ is **perpendicular** to linear decision boundary.
- Datapoints on the "side" with weight vector $\vec{\mathbf{w}}$ are predicted as $+1$; those on the "opposite site" as -1.



Golden retriever: +1, Doberman -1

Ingredients of ML

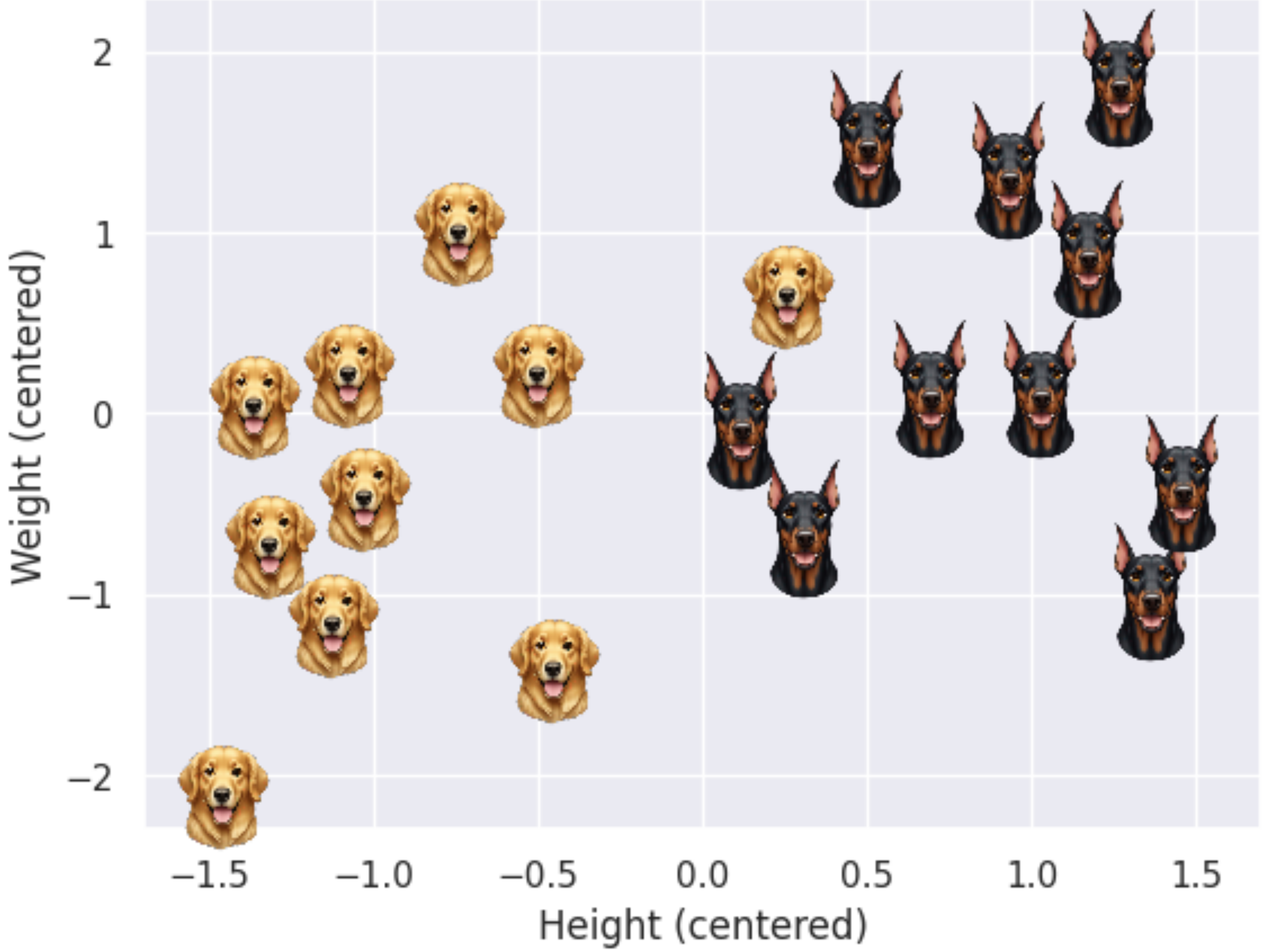
1. **Data.** Height and weight (x_0, x_1)
   and label $y \in \{+1, -1\}$ for Golden Retriever or Doberman.
2. **Model.** Perceptron model.
3. **Model parameters**. Weight vector $\vec{\mathbf{w}} = (w_0, w_1)$.
4. **Training.** Given datapoints, find good model parameters.
5. **Testing.** Evaluate the performance of learned model on new, previously unseen data (i.e. test set).

Perceptron: Training

Initialize all weights to 0.

For each training example with label $y \in \{+1, -1\}$:

- Compute predicted output $\hat{y} = \text{sign}(S)$
- Update weights if incorrect (i.e. $\hat{y} \neq y$)

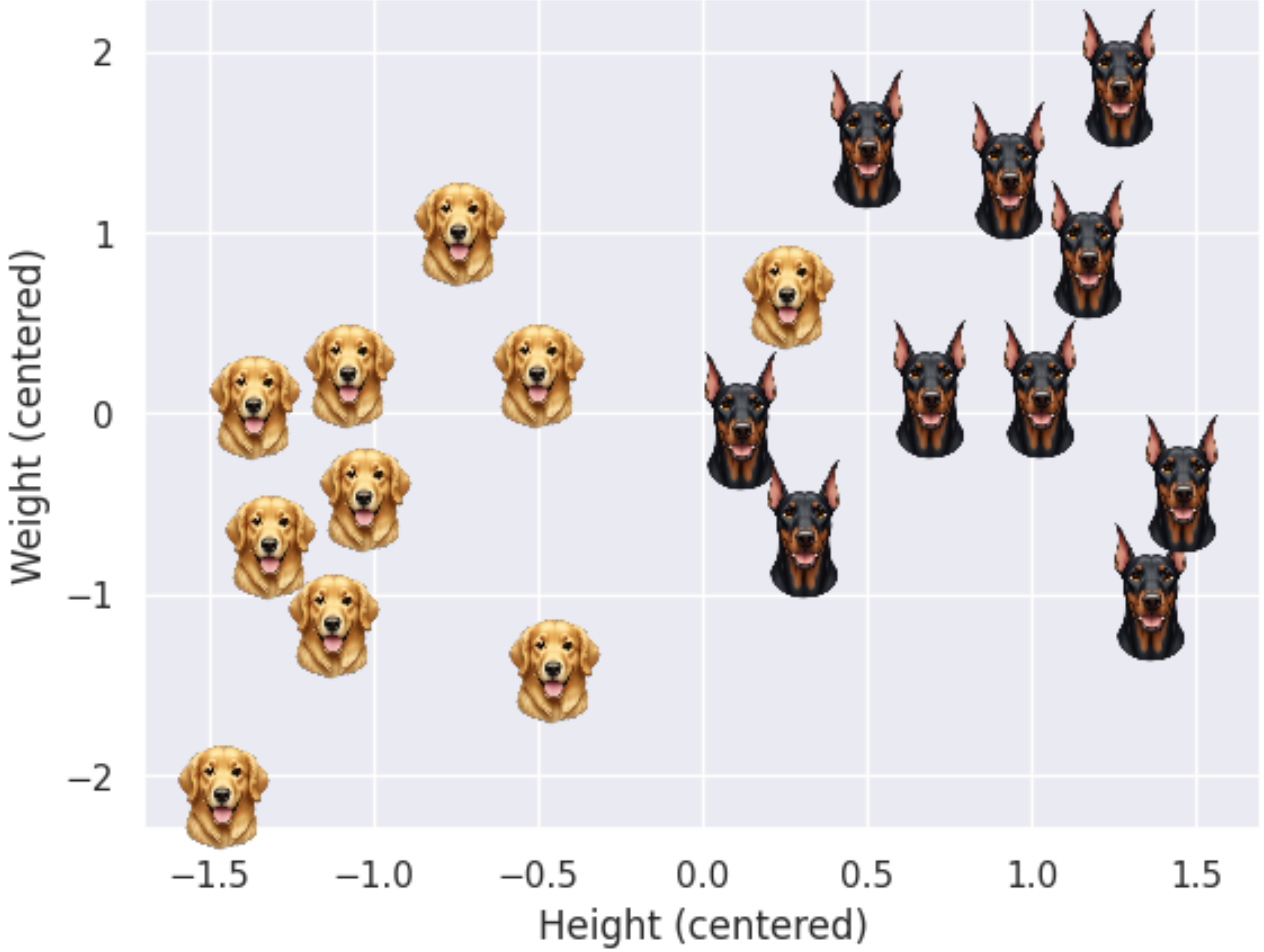


Golden retriever: +1, Doberman -1

Perceptron: Update Rule

For each training example with label $y \in \{+1, -1\}$:

- Compute predicted output
  $S = w_0 x_0 + w_1 x_1$
  $\hat{y} = \text{sign}(S) = +1$ if $S > 0$
  $= -1$ if $S \leq 0$
- If correct ($y = \hat{y}$), do nothing.
- If false positive ($y = -1,\ \hat{y} = +1$):
  $w_j' = w_j - x_j$
- If false negative ($y = +1,\ \hat{y} = -1$):
  $w_j' = w_j + x_j$



Golden retriever: +1, Doberman -1

Poll

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$
- Compute predicted output

$$S = w_0 x_0 + w_1 x_1$$

$$\hat{y} = \text{sign}(S) = +1 \text{ if } S > 0$$

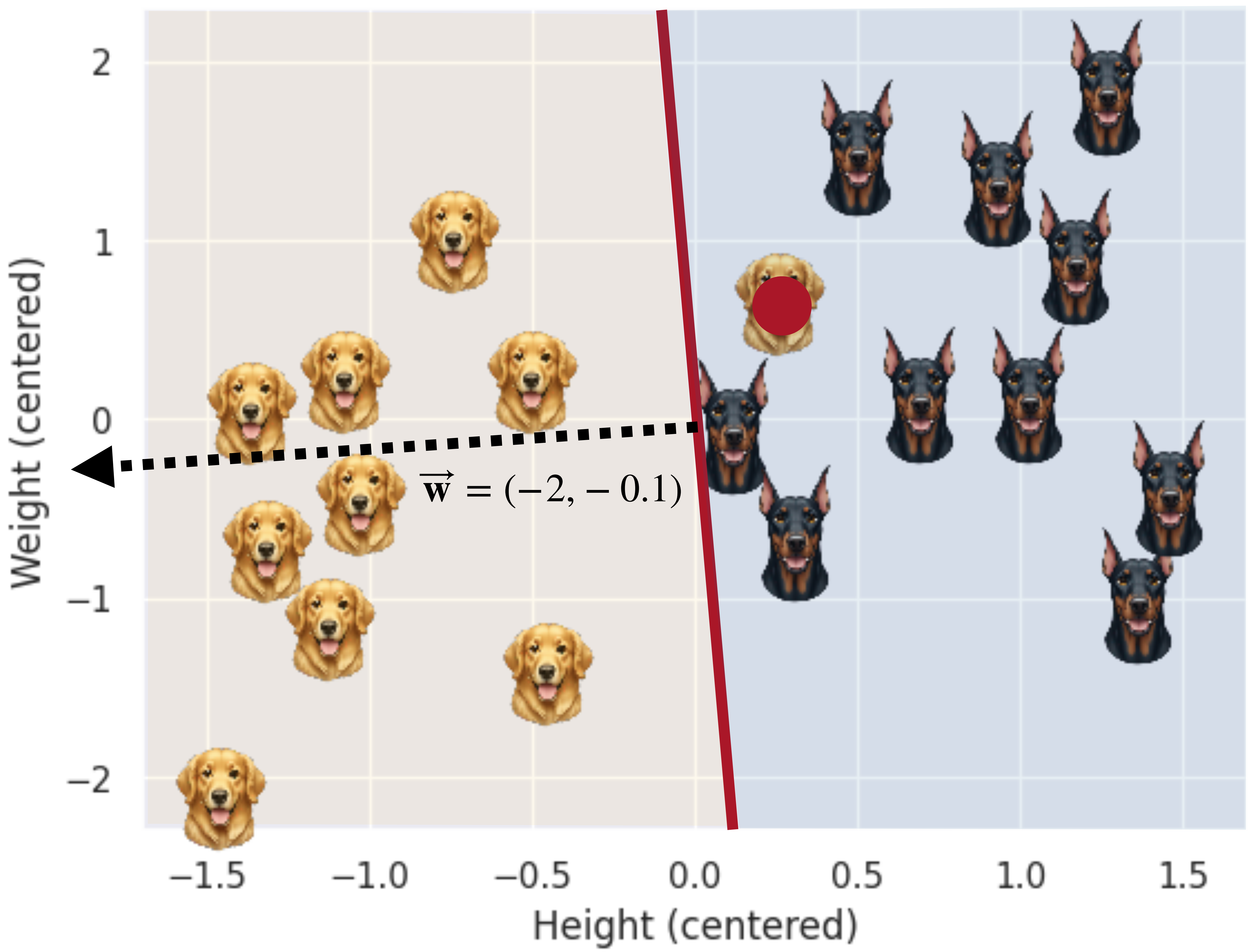
$$= -1 \text{ if } S \leq 0$$

Which scenario is true?

A. Correct prediction ($y = \hat{y}$)

B. False positive ($y = -1,\ \hat{y} = +1$)

C. False negative ($y = +1,\ \hat{y} = -1$)



Golden retriever: +1, Doberman -1

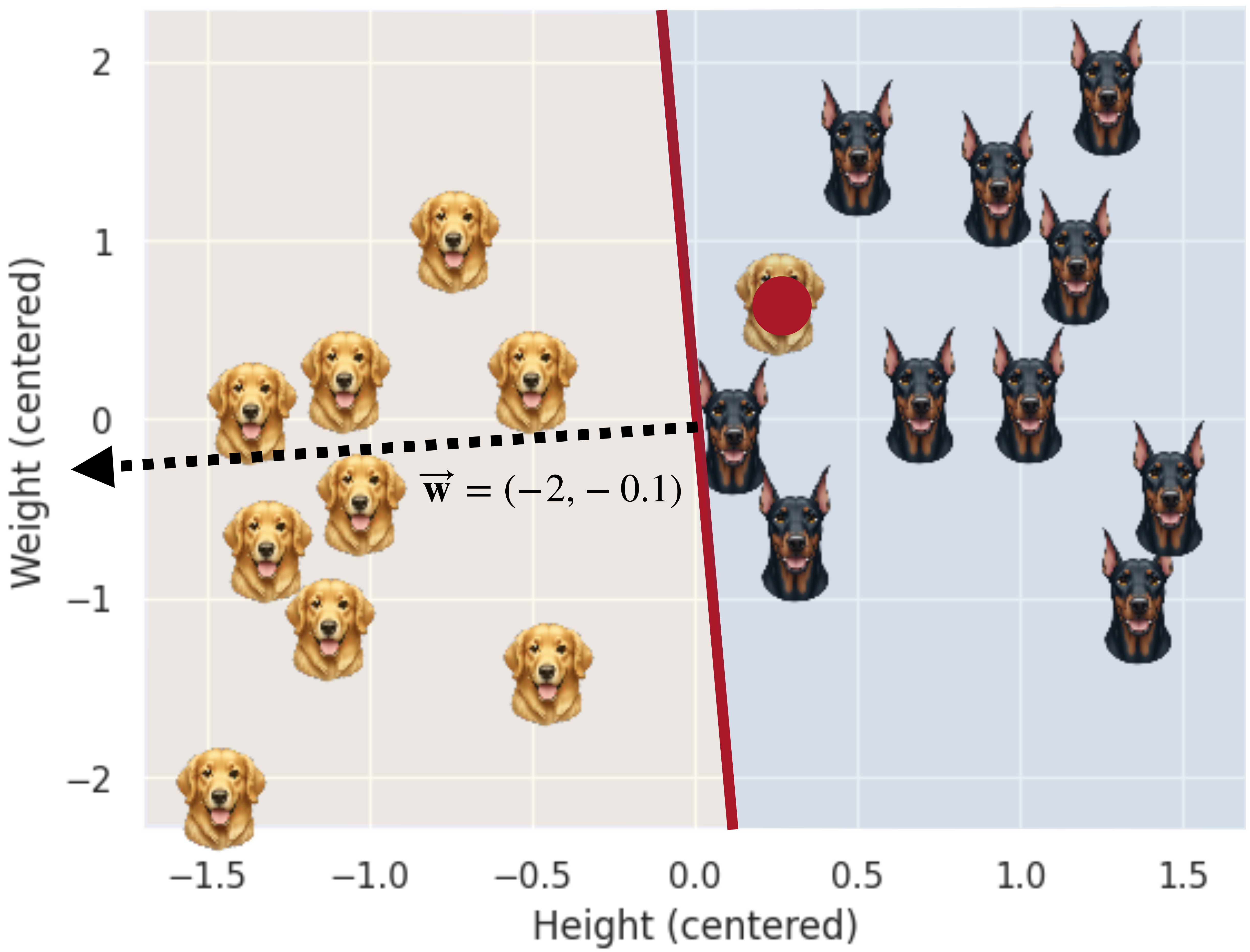
Poll

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$$S = w_0 x_0 + w_1 x_1$$
$$= (-2) \cdot 0.3 + (-0.1) \cdot 0.6$$
$$= -0.6 + (-0.06) = -0.66$$

$$\hat{y} = \text{sign}(S) = -1$$

Which scenario is true?



Golden retriever: +1, Doberman -1

Perceptron: Update Rule

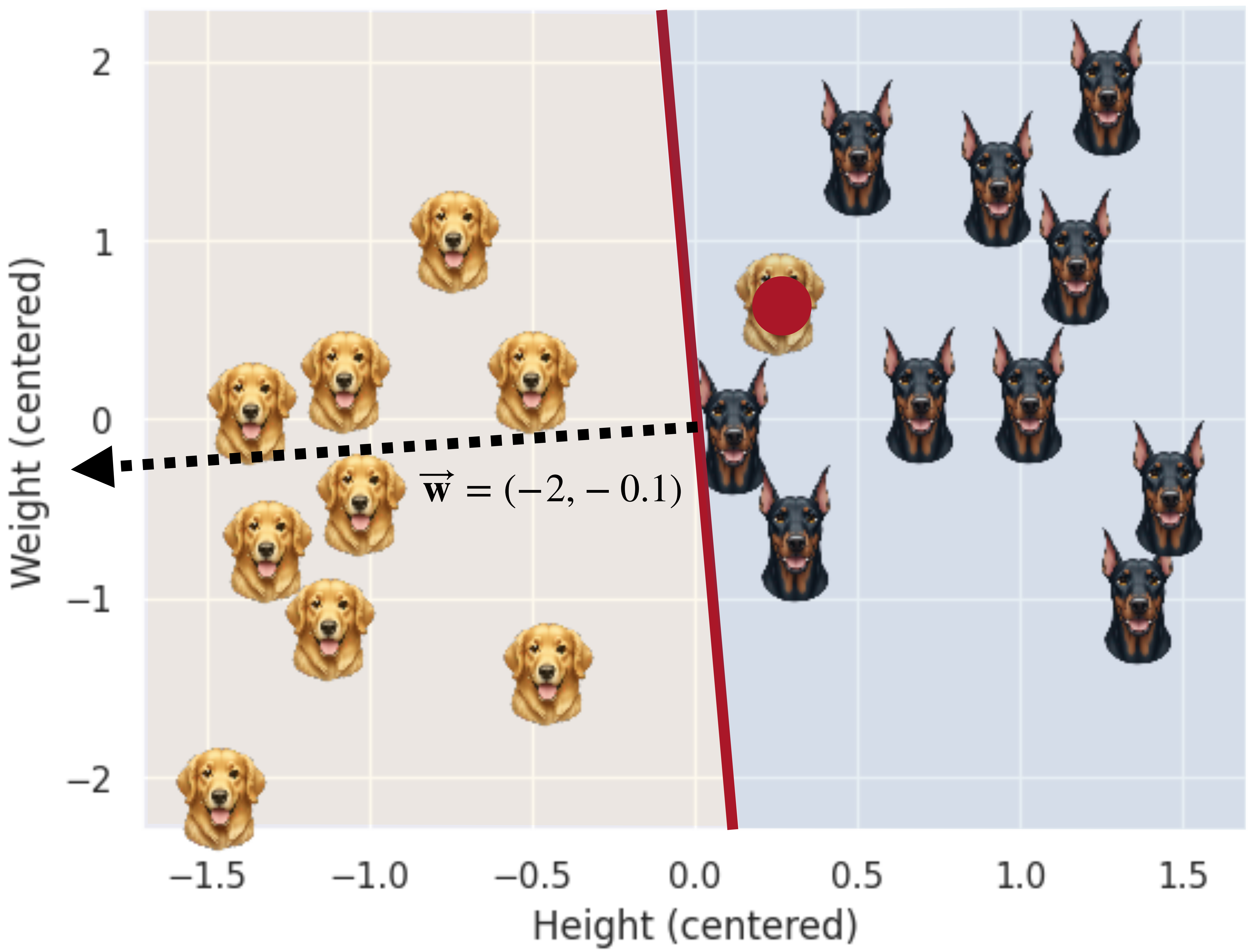
- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$S = w_0 x_0 + w_1 x_1 = -0.66$

$\hat{y} = \text{sign}(S) = -1$

Update rules:

- If correct ($y = \hat{y}$), do nothing.
- If false positive ($y = -1,\ \hat{y} = +1$):
  $w_j' = w_j - x_j$
- If false negative ($y = +1,\ \hat{y} = -1$):
  $w_j' = w_j + x_j$

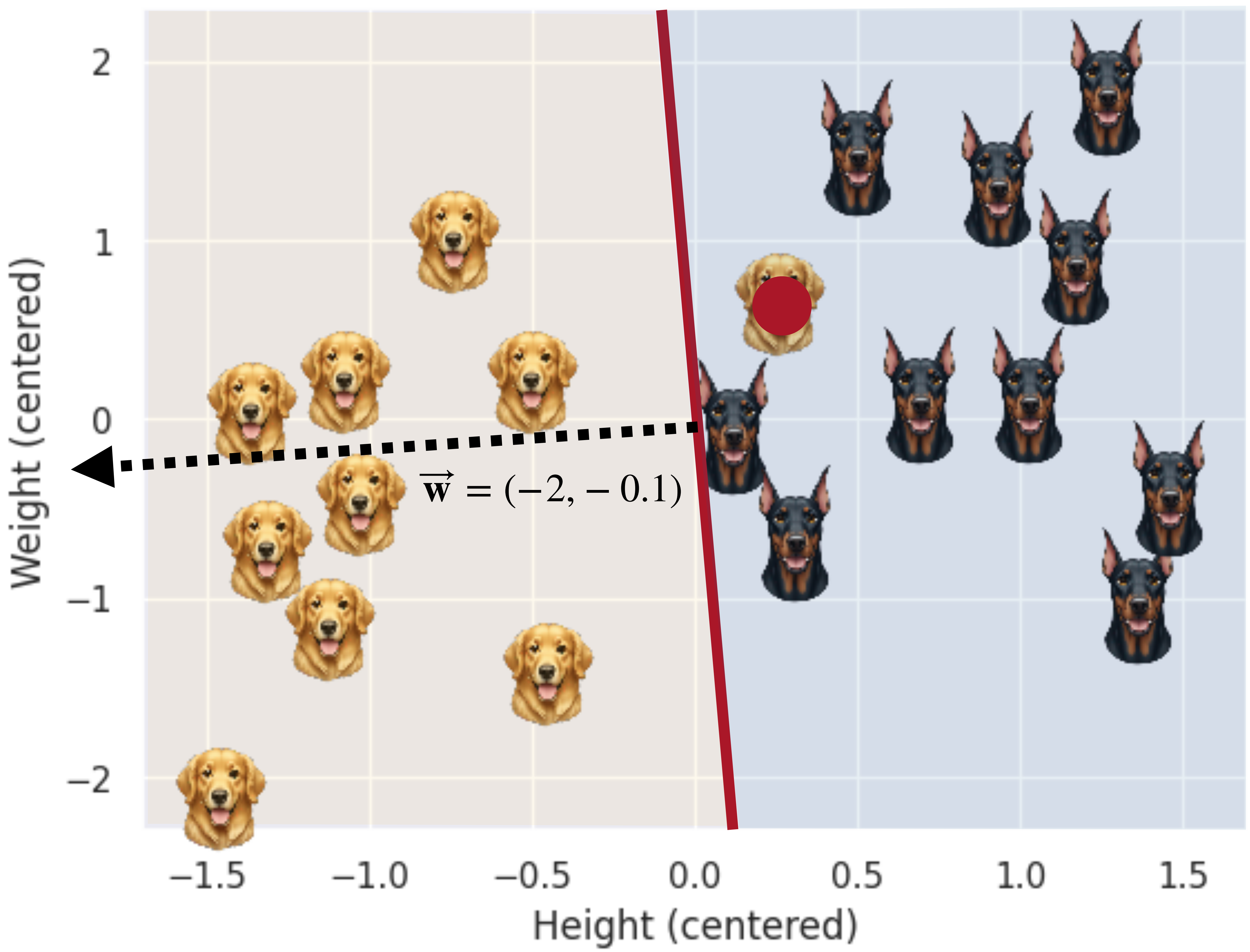


Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$S = w_0 x_0 + w_1 x_1 = -0.66$

$\hat{y} = \text{sign}(S) = -1$



Update rules:

- If correct ($y = \hat{y}$), do nothing.
- If false positive ($y = -1,\ \hat{y} = +1$):
  $w_j' = w_j - x_j$
- **If false negative ($y = +1,\ \hat{y} = -1$):**
  $w_j' = w_j + x_j$

Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Weights: $w_0 = -2,\ w_1 = -0.1$

$S = w_0x_0 + w_1x_1 = -0.66$

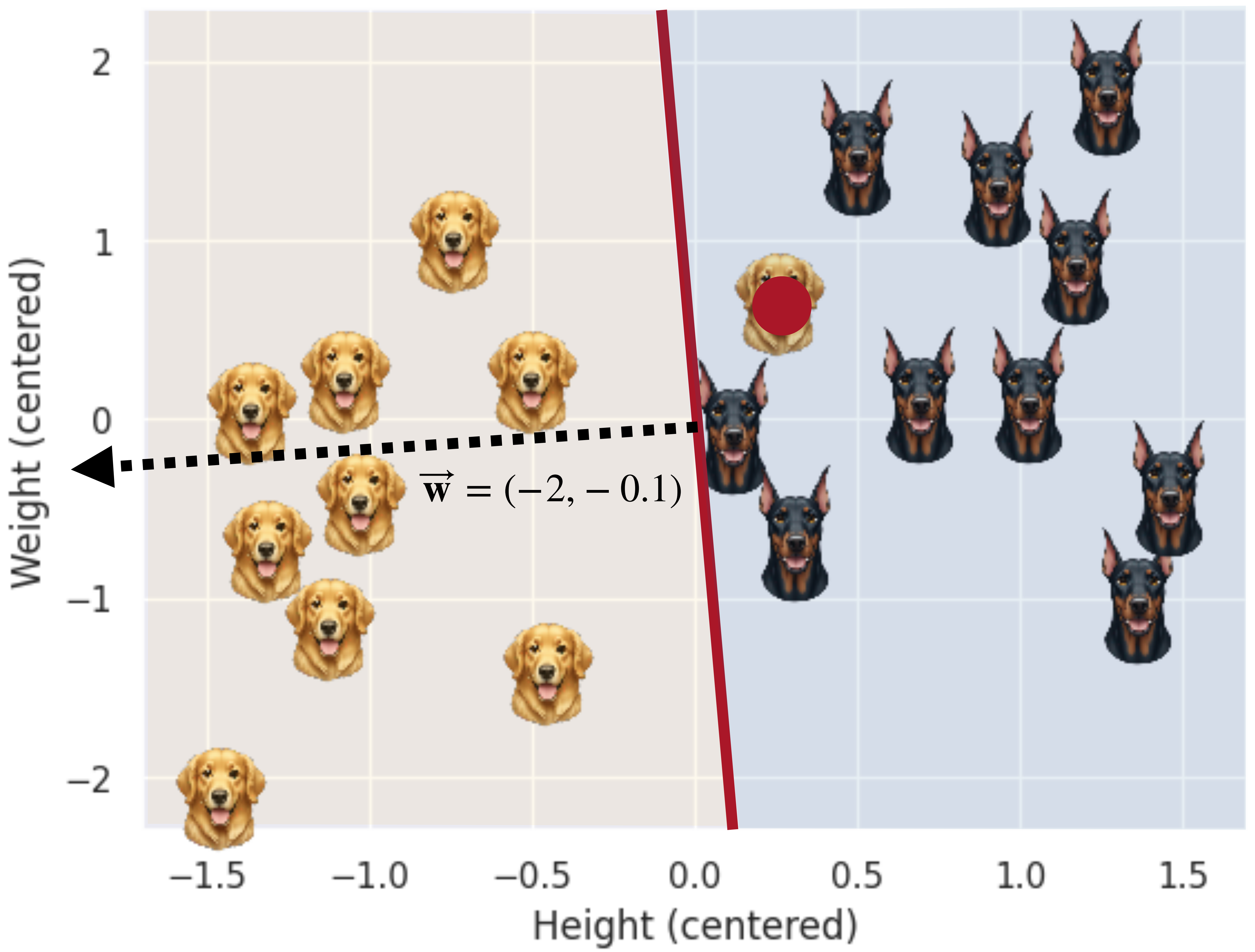
$\hat{y} = \text{sign}(S) = -1$

If false negative ($y = +1,\ \hat{y} = -1$):

$w_j' = w_j + x_j$

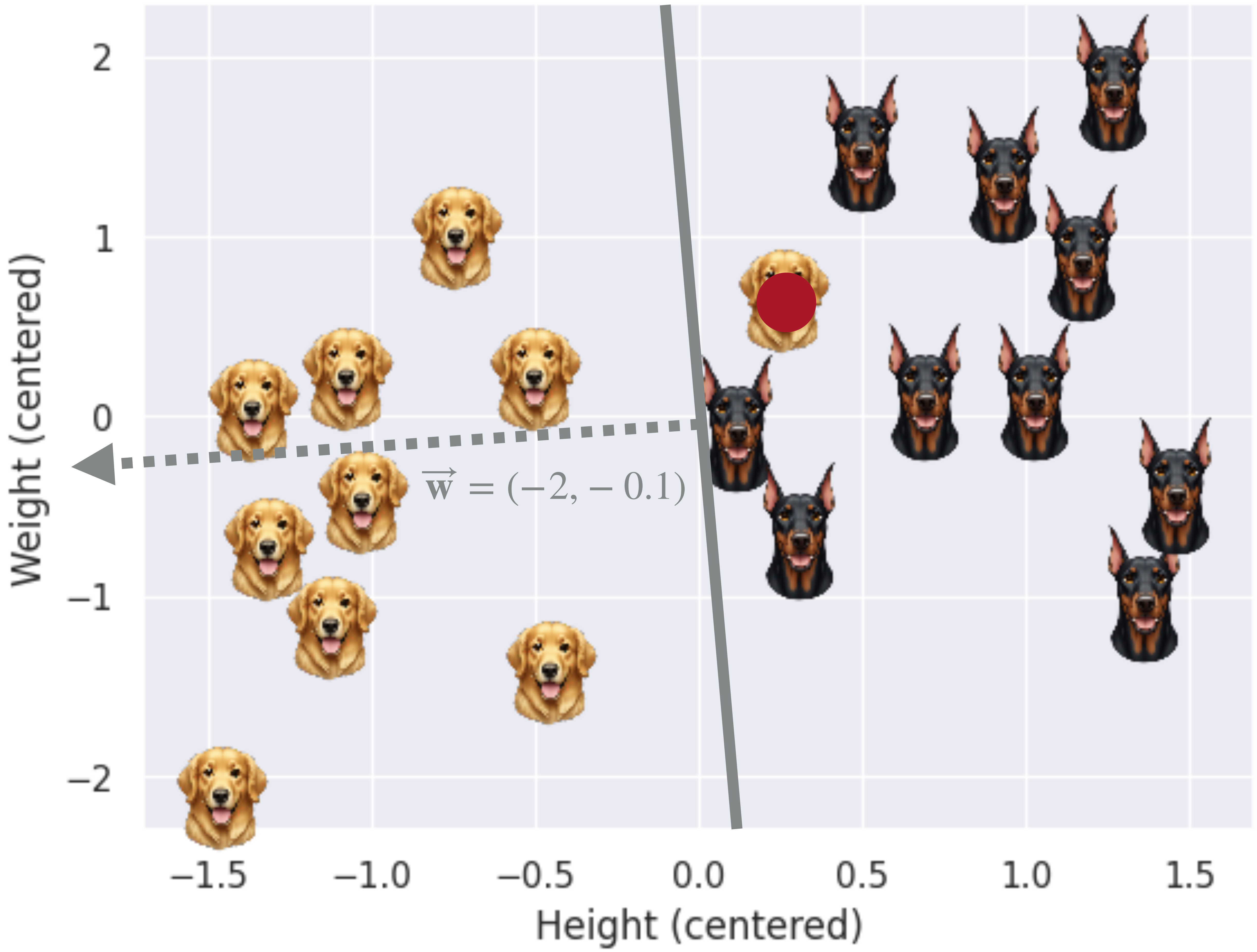
$w_0' = w_0 + x_0 = -2 + 0.3 = -1.7$

$w_1' = w_1 + x_1 = -0.1 + 0.6 = 0.5$



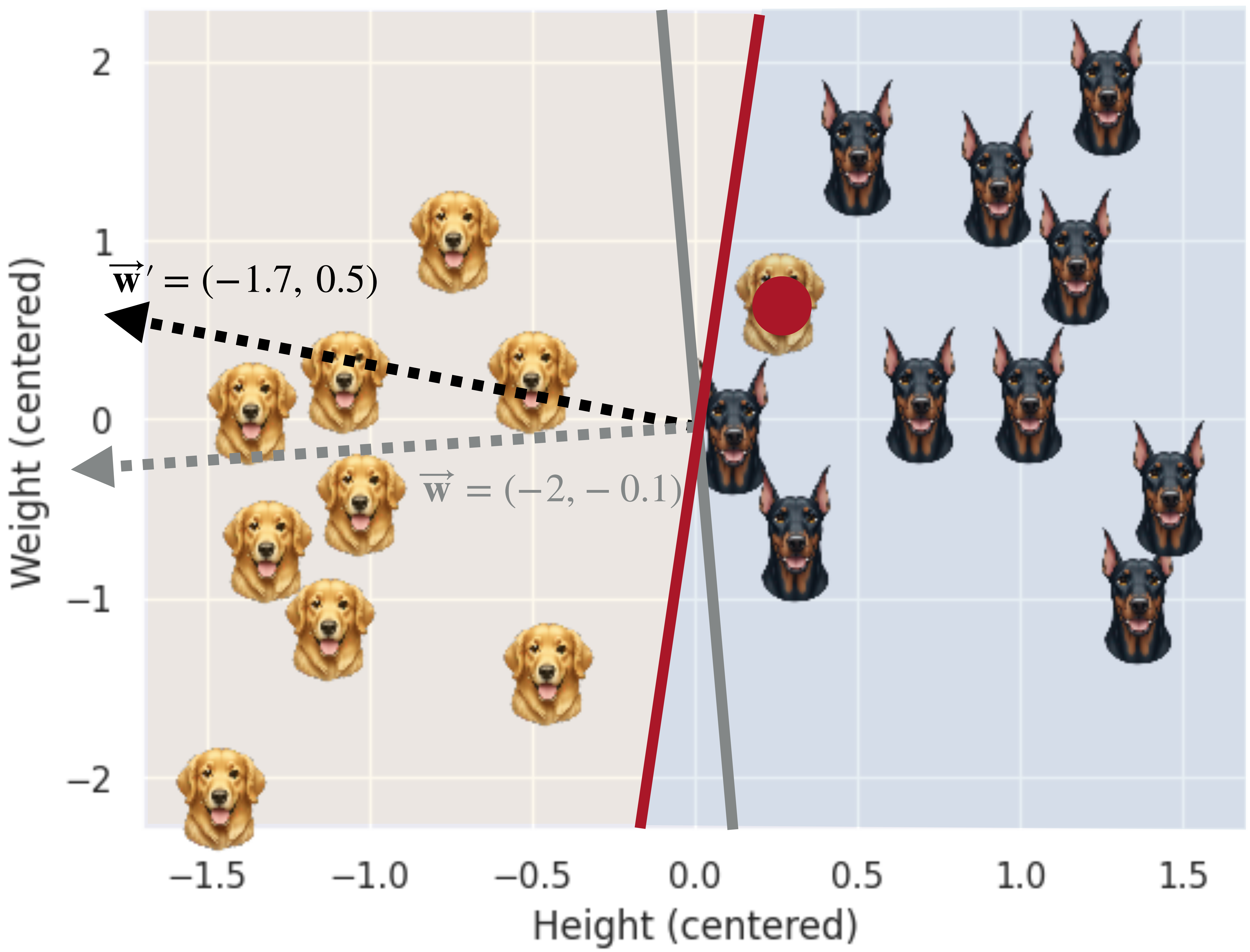
Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Old weights: $w_0 = -2,\ w_1 = -0.1$



Perceptron: Update Rule

- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- Old weights: $w_0 = -2,\ w_1 = -0.1$
- New weights: $w_0' = -1.7,\ w_1' = 0.5$



Perceptron: Update Rule

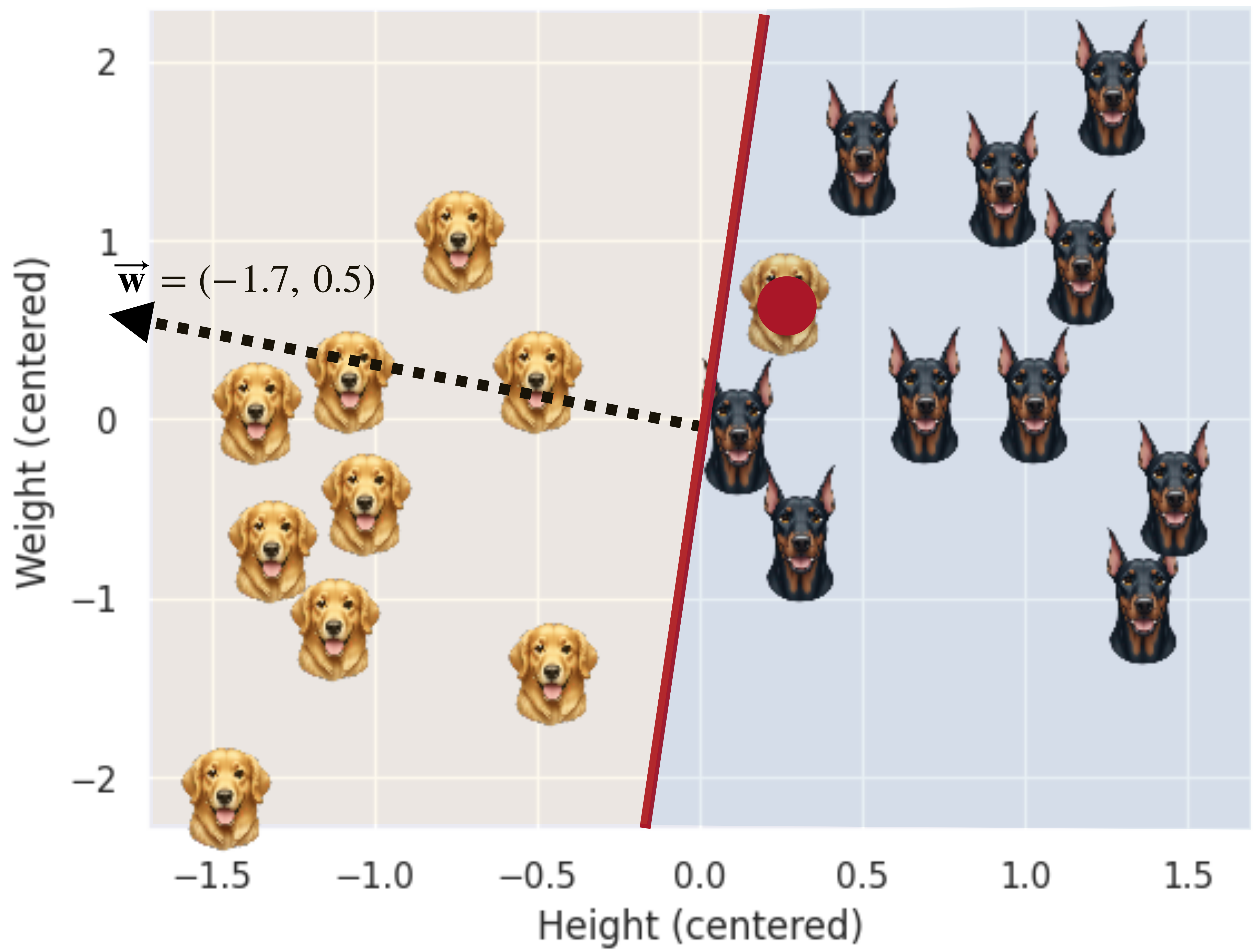
- Inputs: $x_0 = 0.3,\ x_1 = 0.6,\ y = +1$
- New weights: $w_0 = -1.7,\ w_1 = 0.5$

$S = w_0 x_0 + w_1 x_1 = -0.21$

$\hat{y} = \text{sign}(S) = -1$

Still false negative, but closer to boundary.

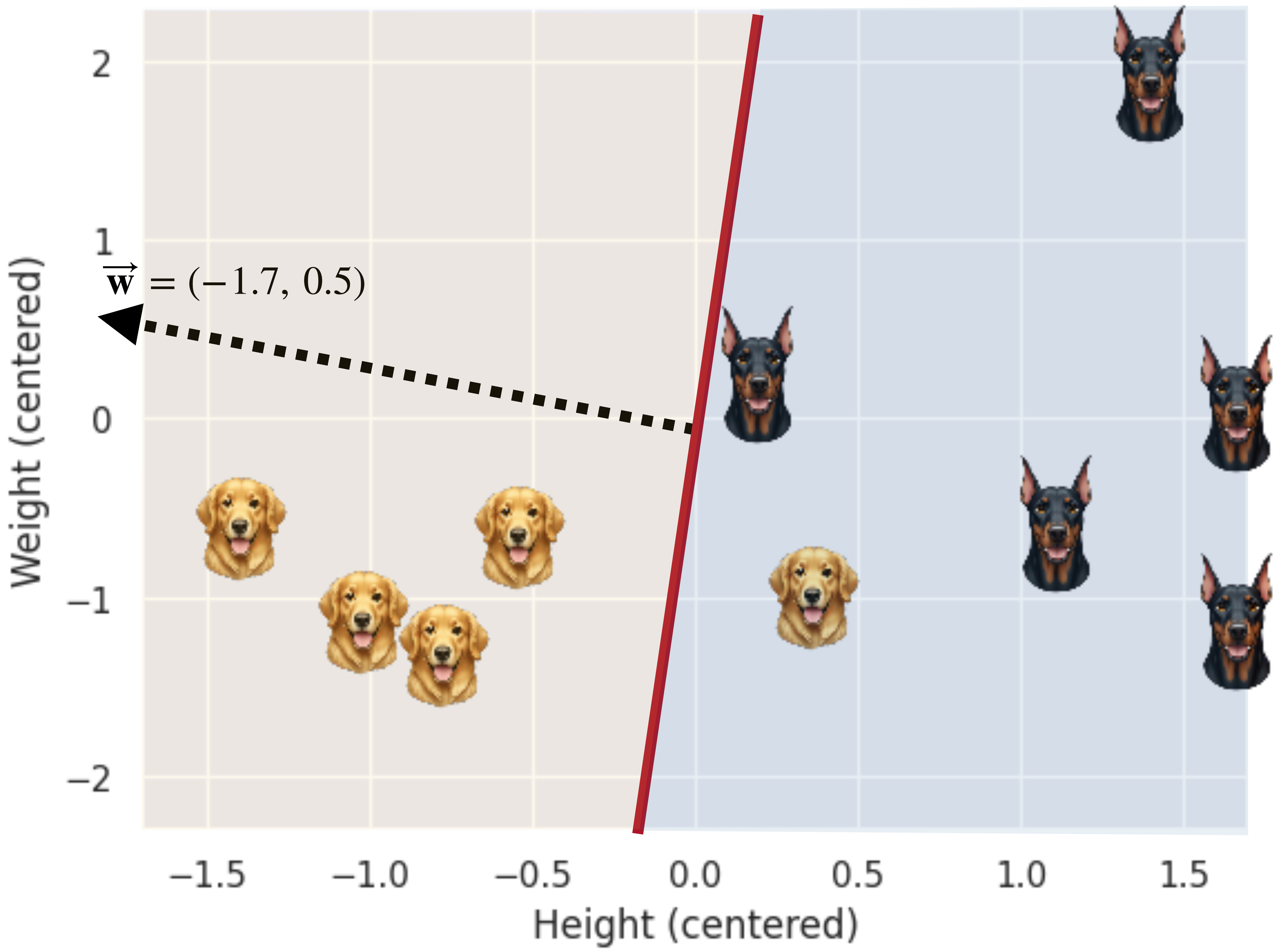
Intuition: Update rule moves decision boundary to be better for the current training example.



Perceptron: Testing

Evaluate performance using an evaluation metric on an unseen set of examples (e.g. test examples).

$$\text{Error rate} = \frac{\#\ \text{incorrect}}{\#\ \text{total}}$$



Poll

Evaluate performance using an evaluation metric on an unseen set of examples (e.g. test examples).

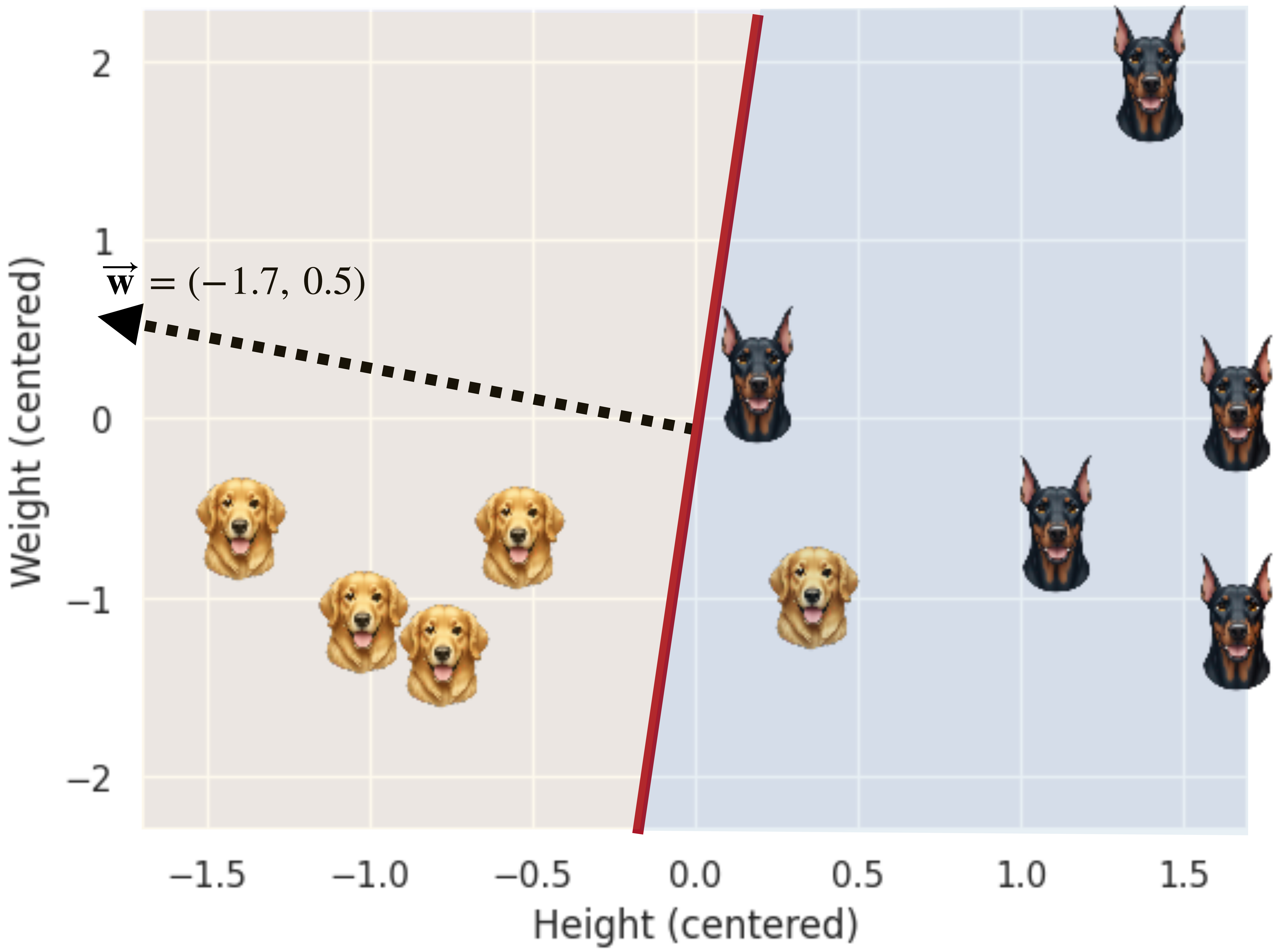
$$\text{Error rate} = \frac{\#\text{ incorrect}}{\#\text{ total}}$$

Consider the following unseen examples (5 Golden Retrievers, 5 Dobermans) and the given Perceptron model.

What is the error rate on test set?

A. $0/10$ C. $2/10$

B. $1/10$ D. $9/10$



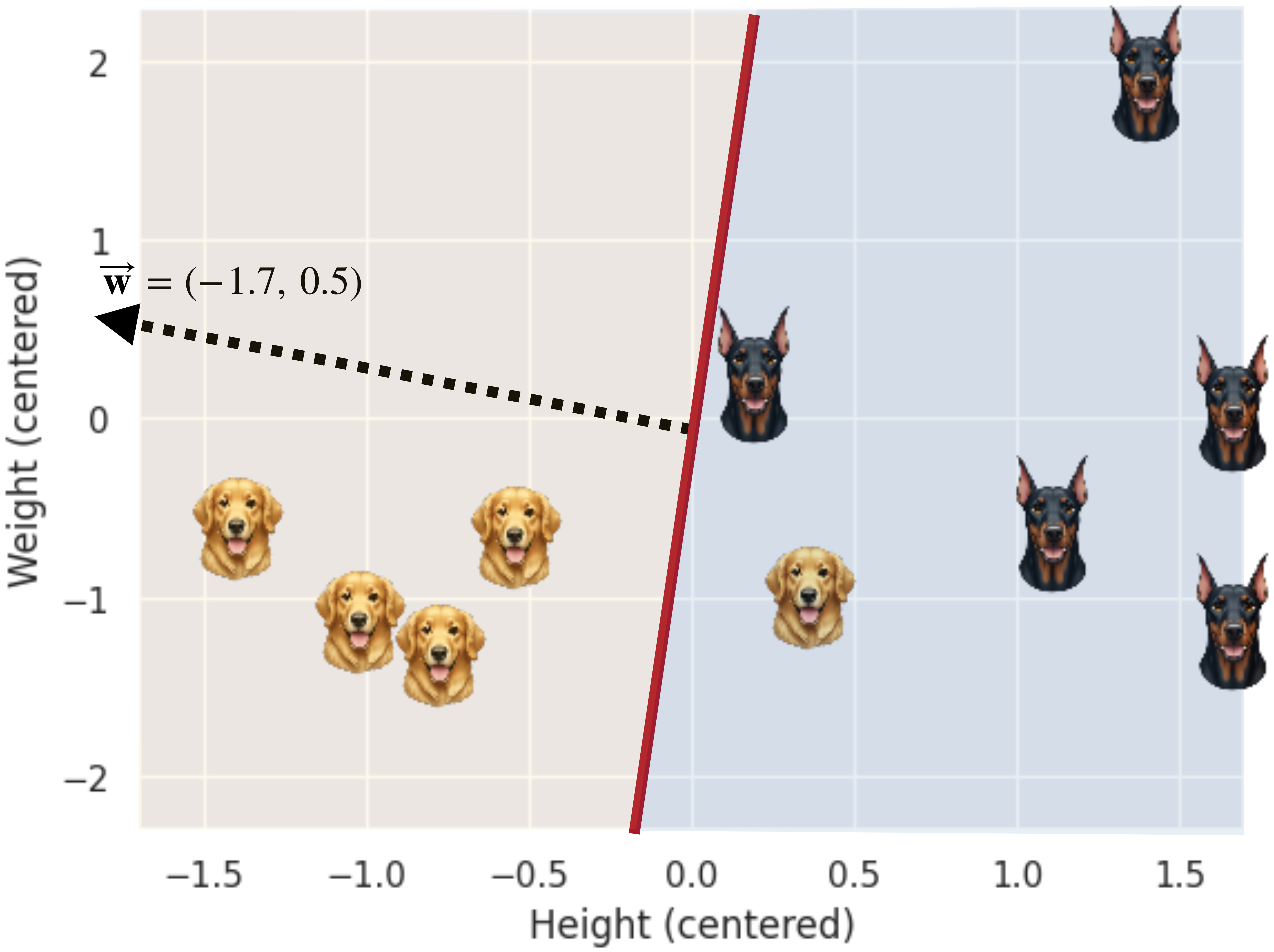
Poll

Evaluate performance using an evaluation metric on an unseen set of examples (e.g. test examples).

$$\text{Error rate} = \frac{\#\ \text{incorrect}}{\#\ \text{total}}$$

Consider the following unseen examples (5 Golden Retrievers, 5 Dobermans) and the given Perceptron model.

What is the error rate on test set?

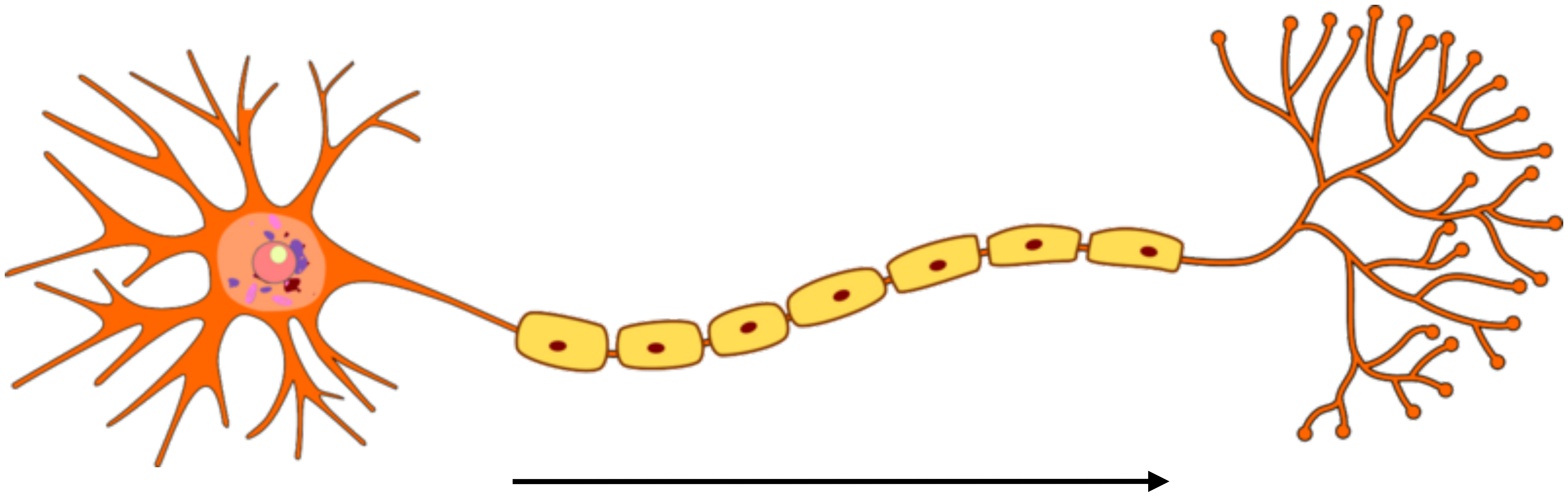


Ingredients of ML

1. **Data.** Height and weight (x_0, x_1) and label $y \in \{+1, -1\}$ for Golden Retriever or Doberman.
2. **Model.** Perceptron model.
3. **Model parameters**. Weight vector $\vec{\mathbf{w}} = (w_0, w_1)$.
4. **Training.** Use the Perceptron update rule to train on training examples.
5. **Testing.** Compute error rate on new, previously unseen data (i.e. test set).

Multi-layer Perceptron

Biological Neuron



Dendrites

Receive information from other neurons

Action potential

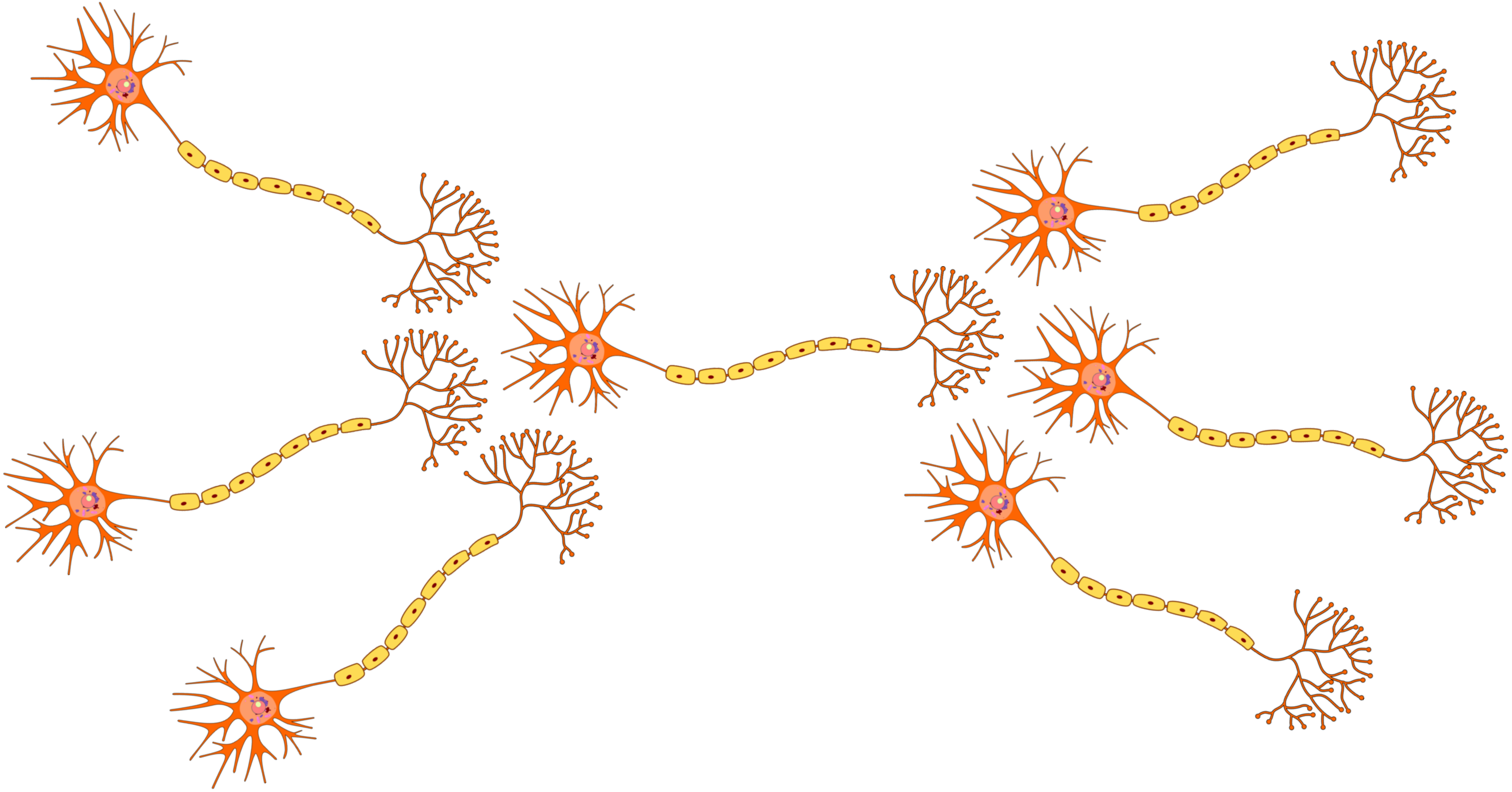
Neuron "fires" based on received information

Synaptic terminals

Relay "firing" to other neurons

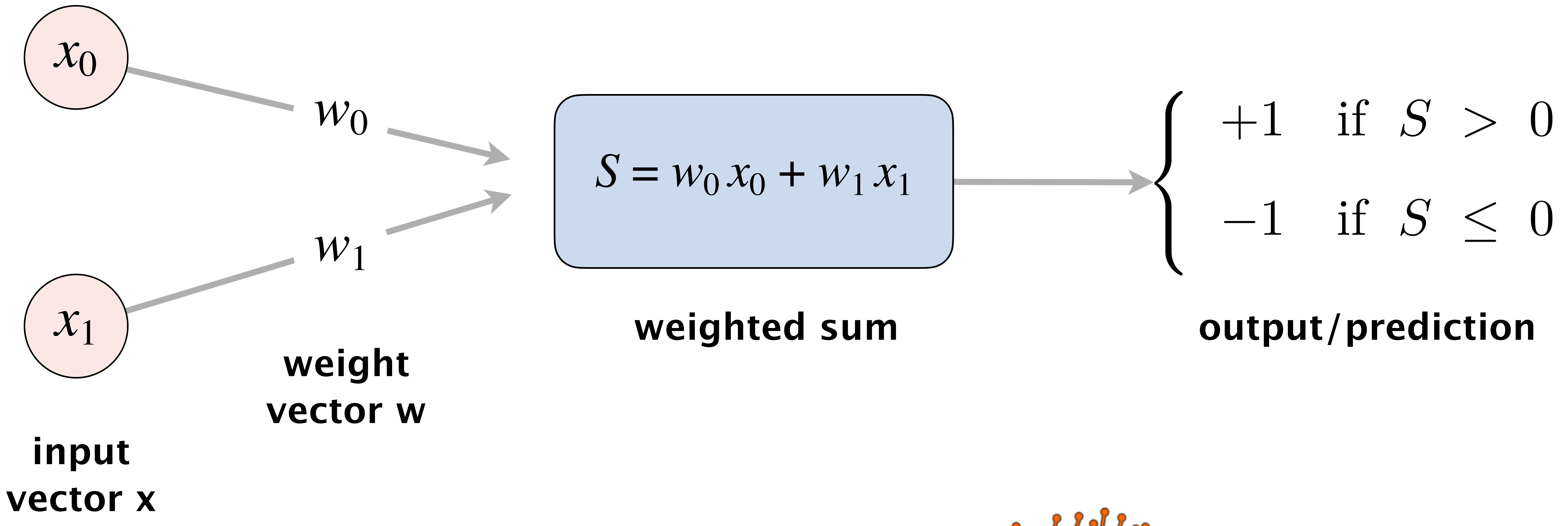
Neuron firing (or not) is a binary output.

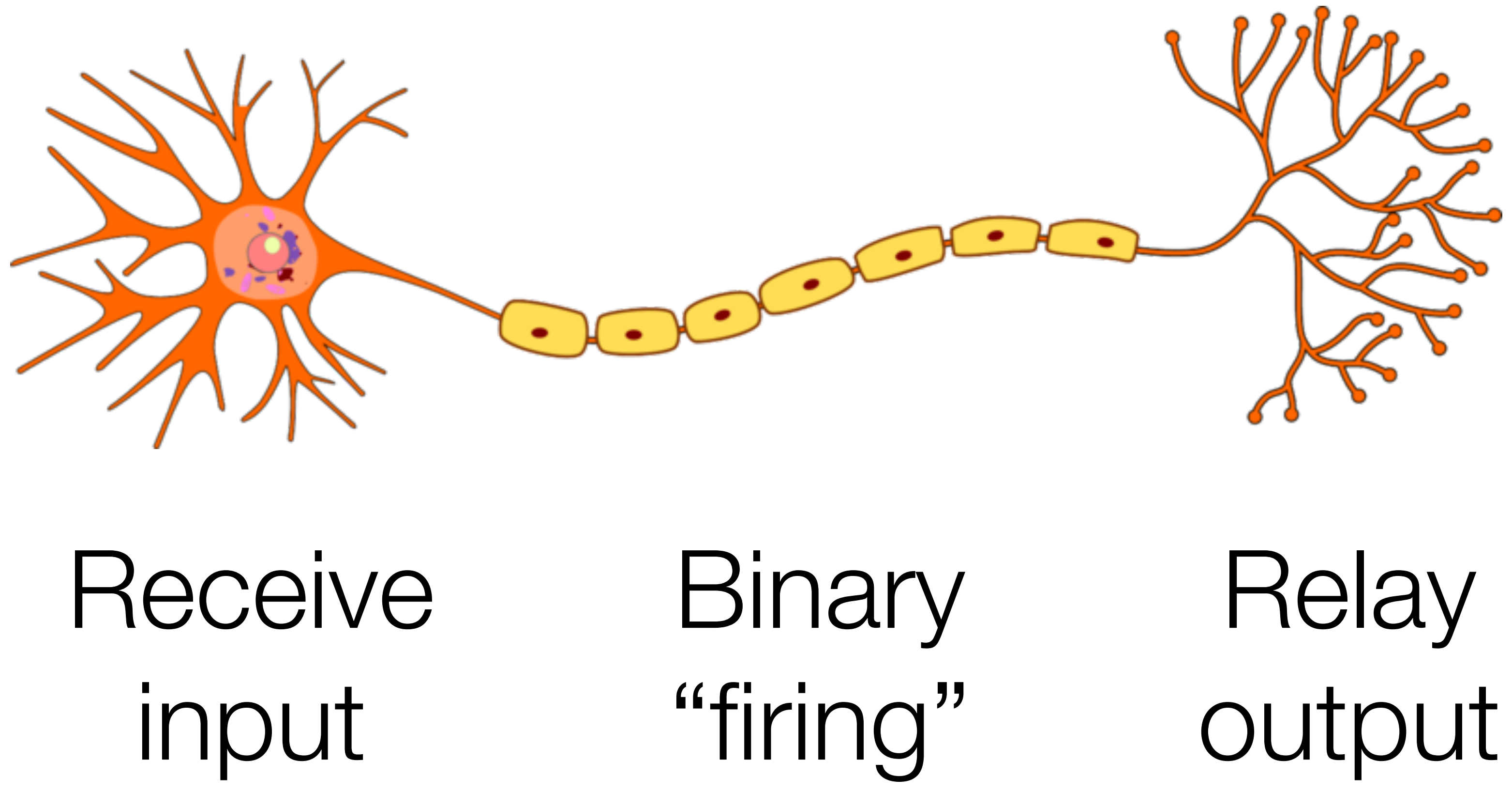
Biological Neuron



Perceptron

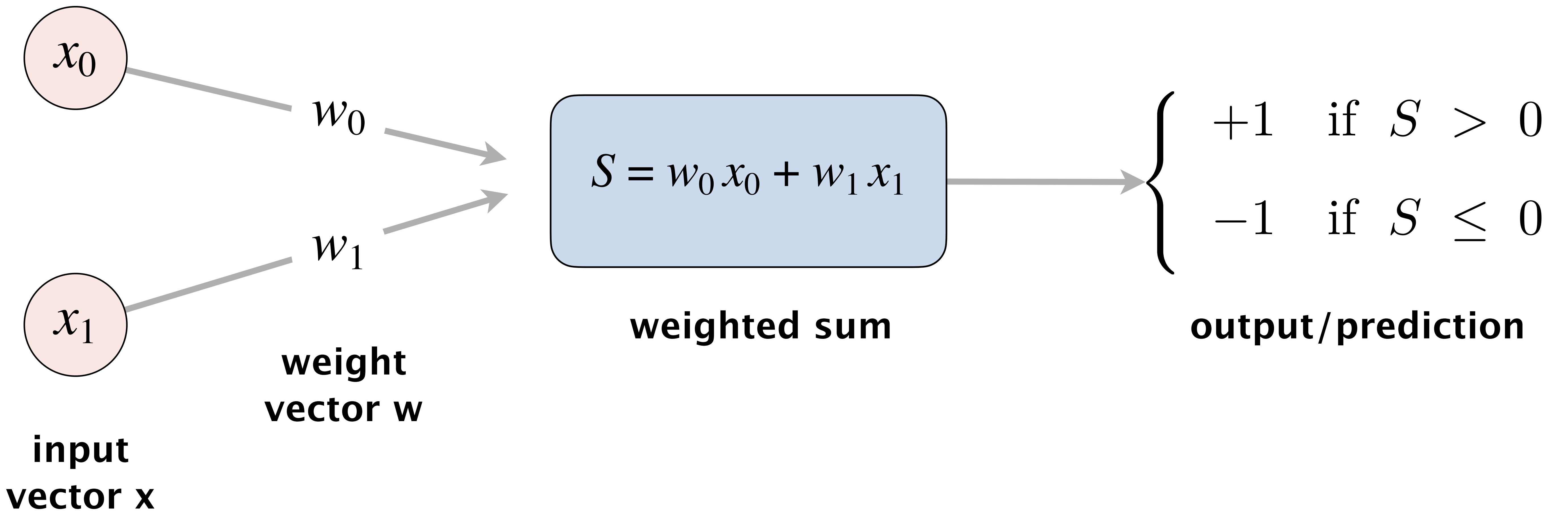
Takes several inputs and produces a single binary output.

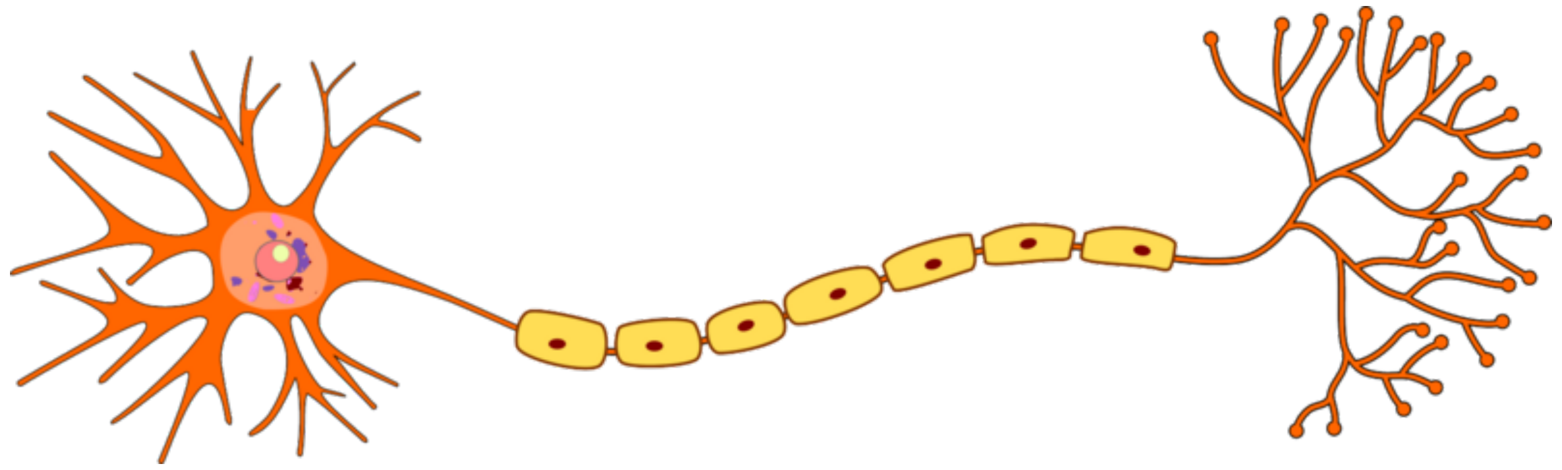


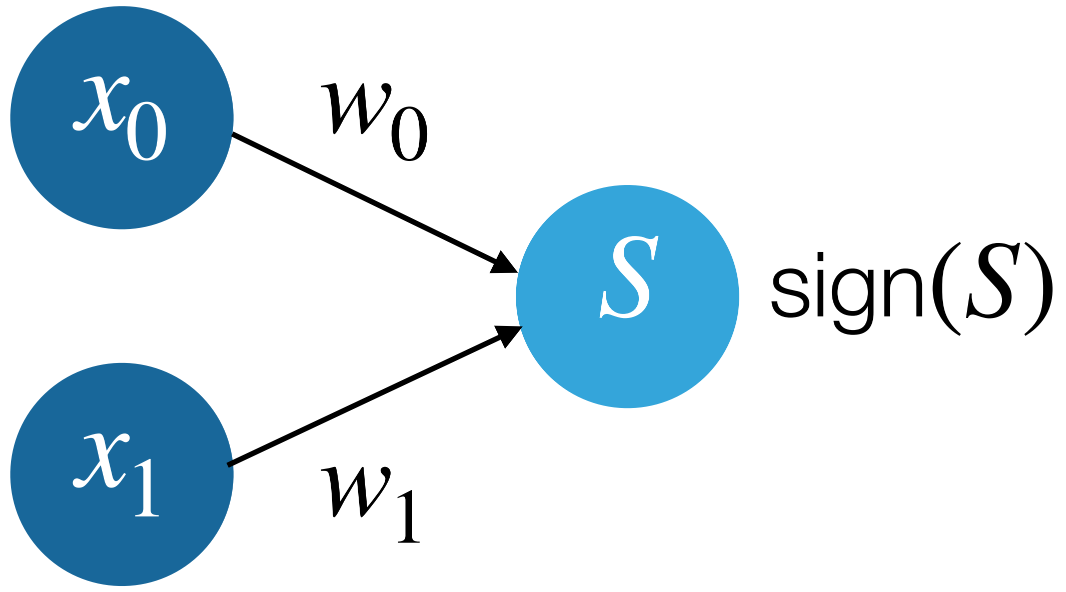


Perceptron

Takes several inputs and produces a single binary output.



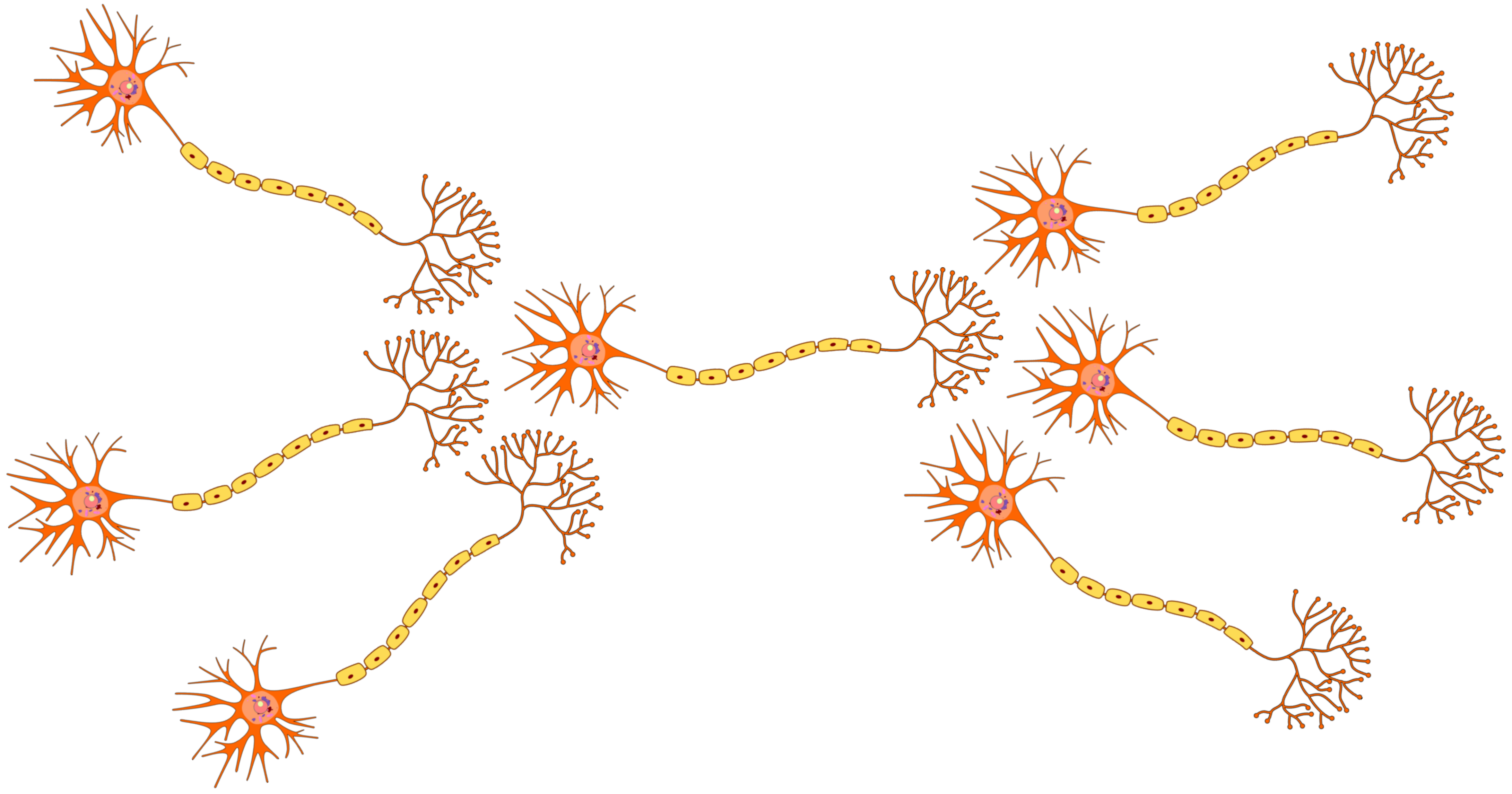


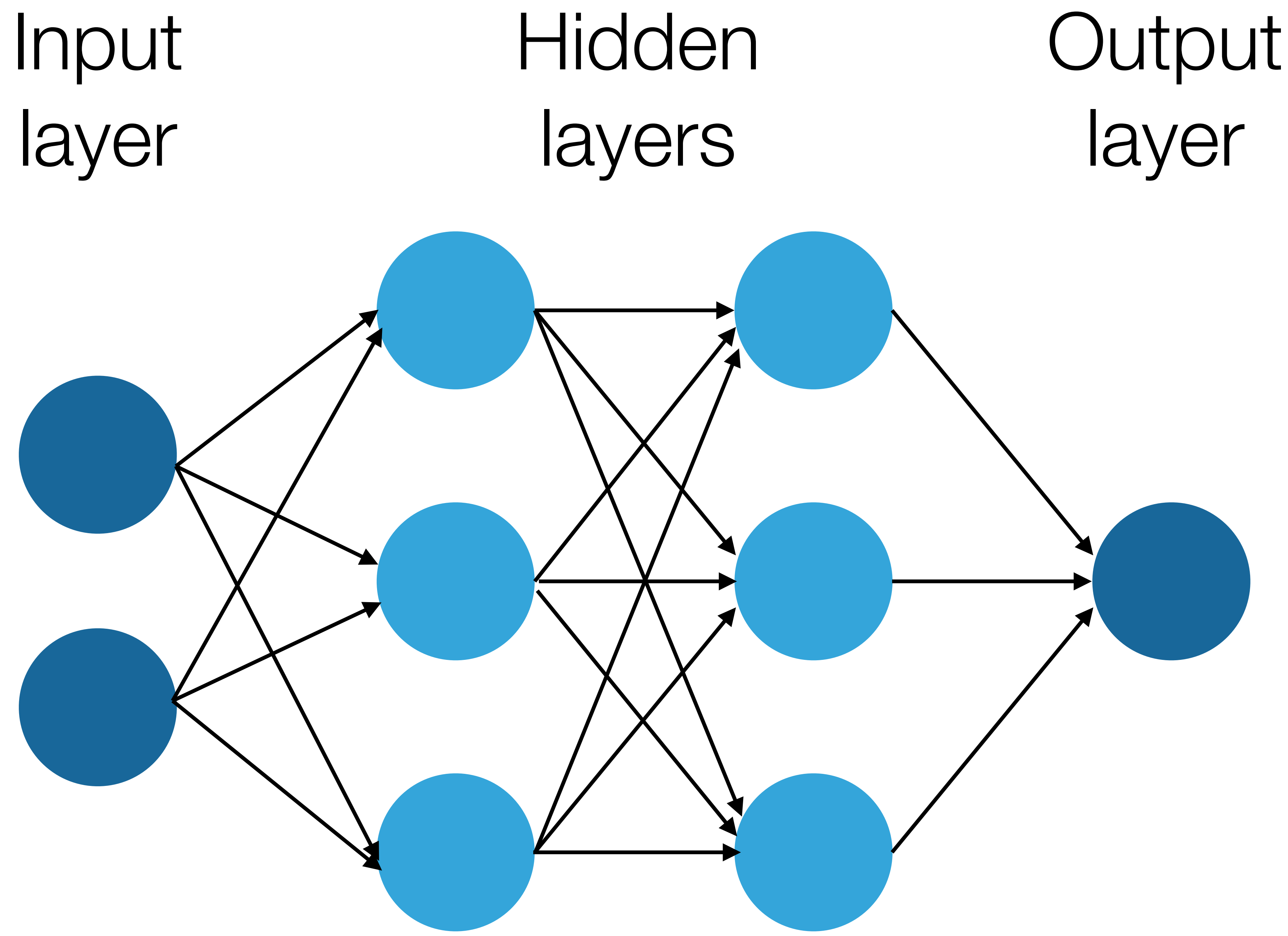


Multi-Layer Perceptron (MLP)

Composed of multiple layers of perceptron neurons, where each neuron is connected to all neurons in the next layer (i.e. fully-connected).

MLPs are an example of a deep neural network.



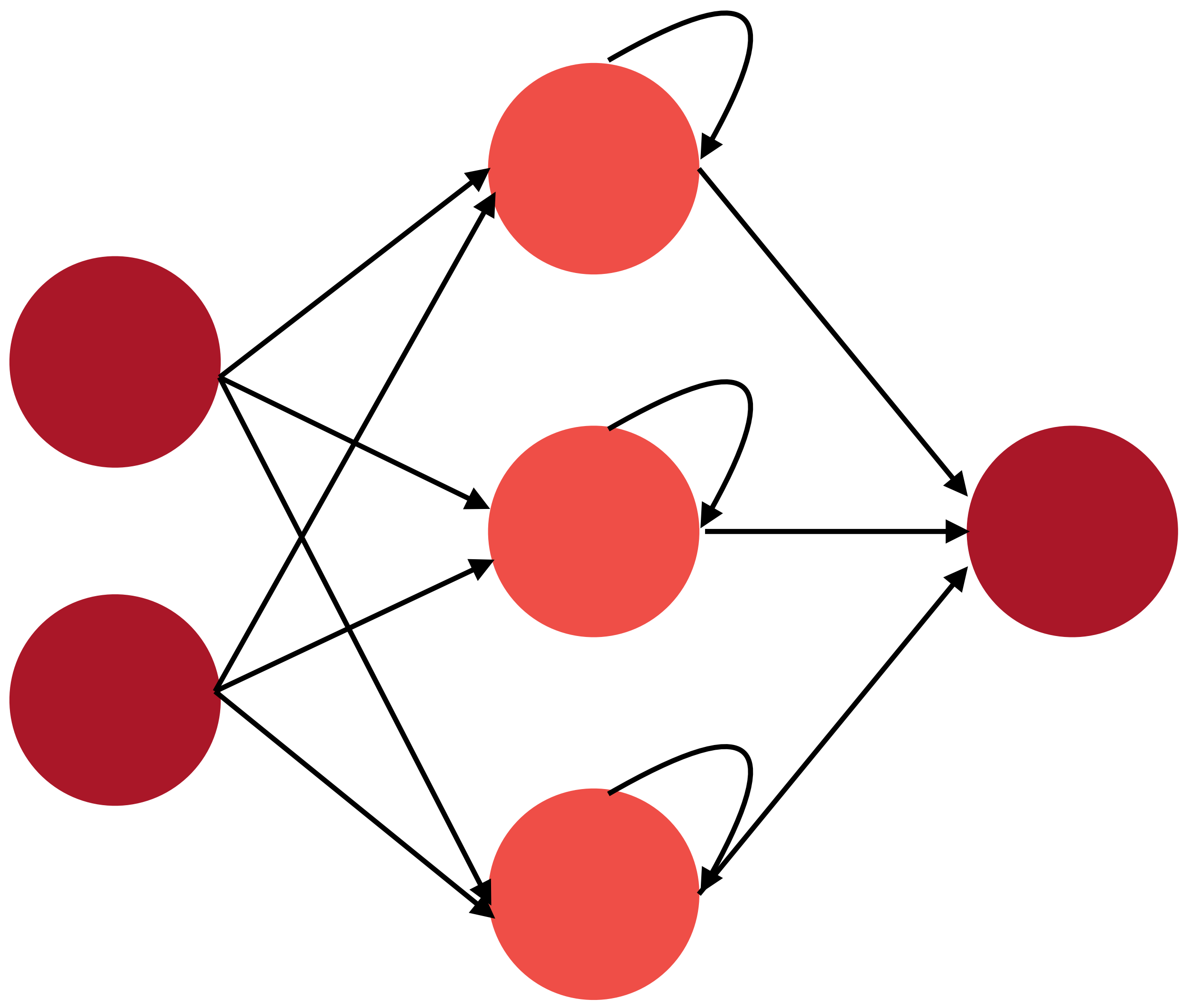


Deep Learning

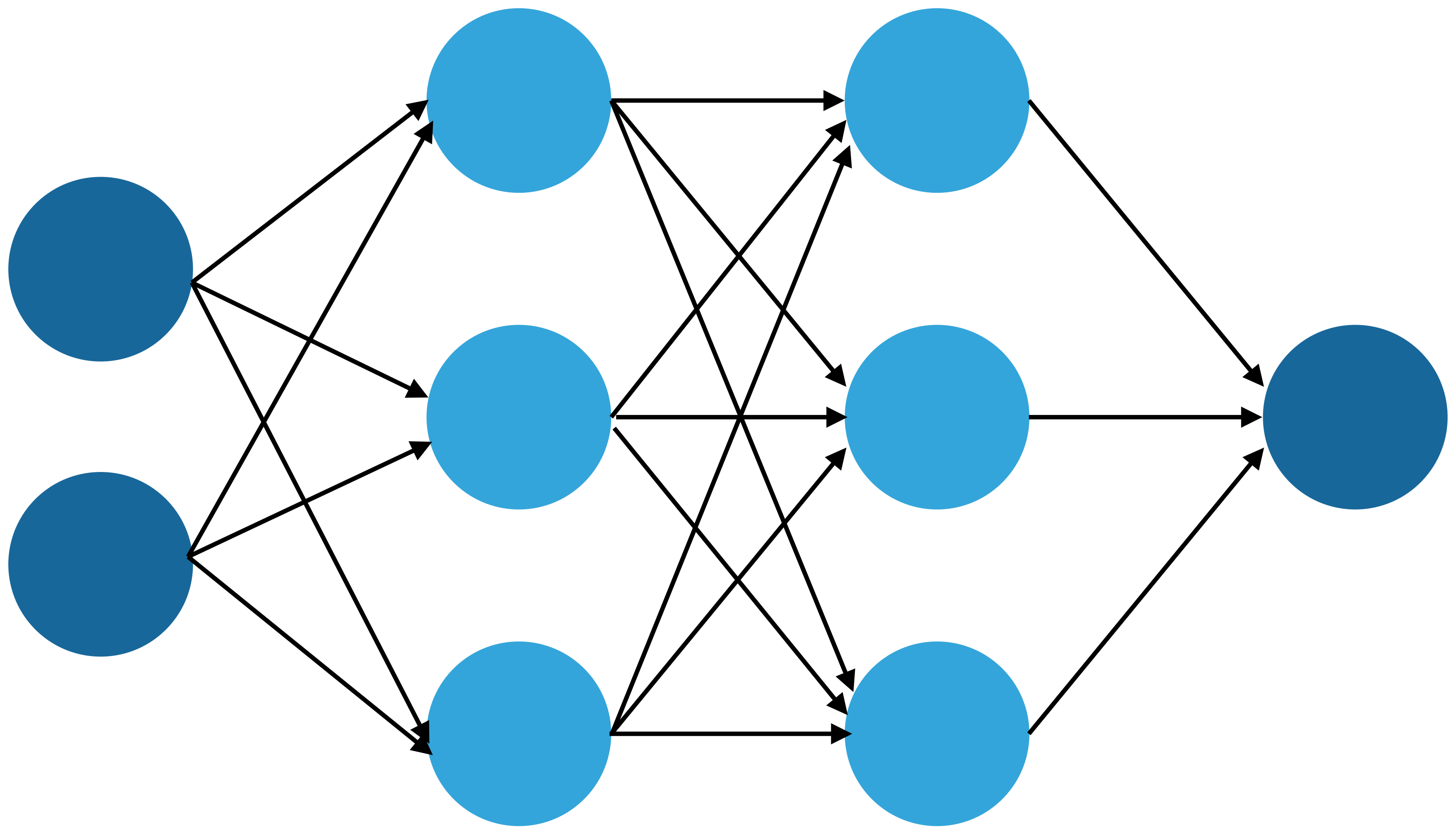
Refers to ML models that use deep neural networks.

Popular neural network architectures:

- Recurrent neural networks
- Convolutional neural networks
- Transformers



Recurrent NN



Fully-connected NN

Multi-class Perceptron

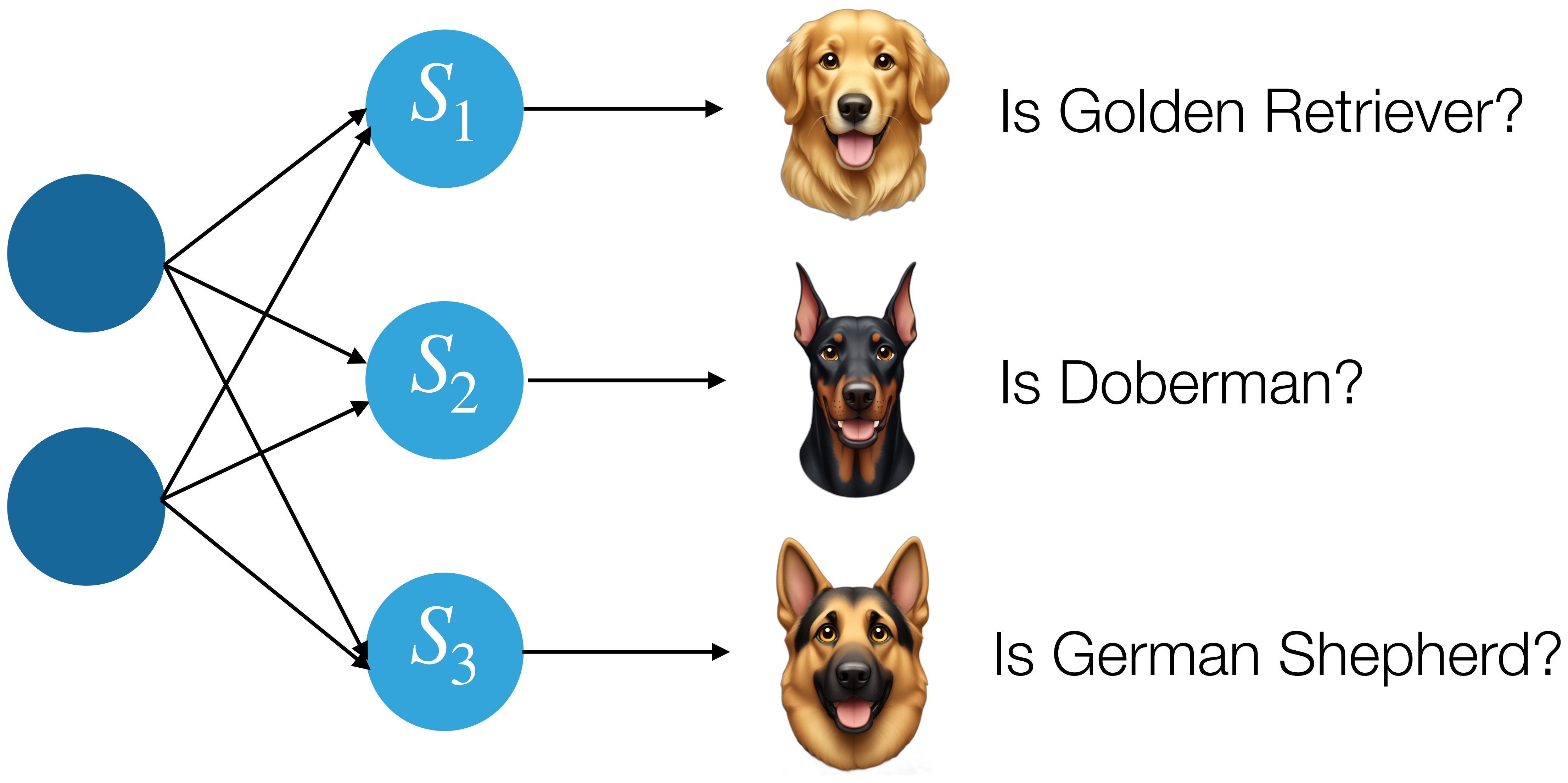
Multi-class Classification

Predict one of **k** classes (k > 2).

Suppose we have **k = 3** dog species we are trying to classify.

Train **k = 3** binary Perceptron models and compute **3** weighted sums: S_1, S_2, S_3.

Final prediction is the class with the largest weighted sum.



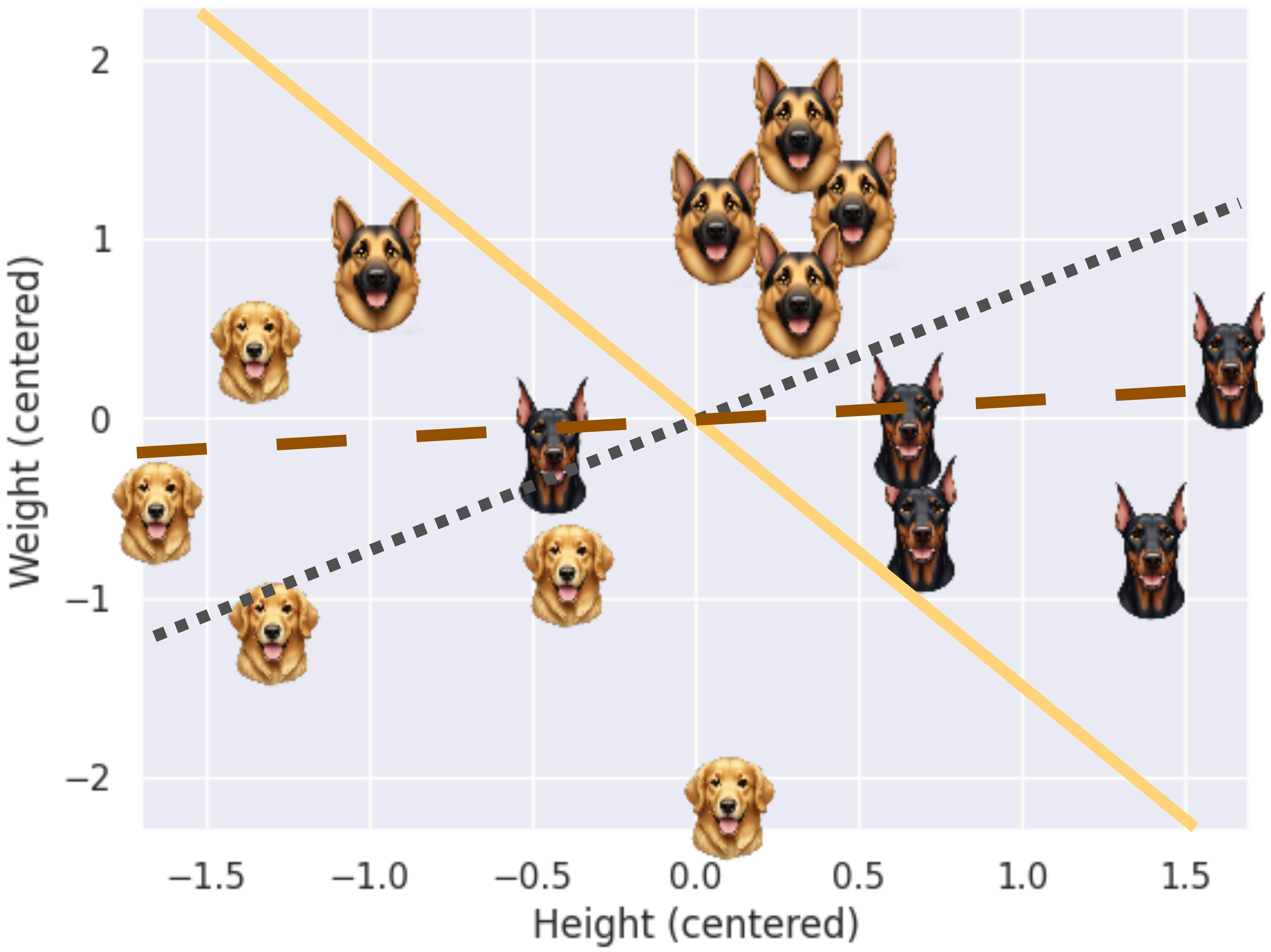
Multi-class Classification: Geometric Perspective

Predict one of **k** classes (k > 2).

Suppose we have **k = 3** dog species we are trying to classify.

Train **k = 3** binary Perceptron models and compute **3** weighted sums: S_1, S_2, S_3.

Final prediction is the class with the largest weighted sum.



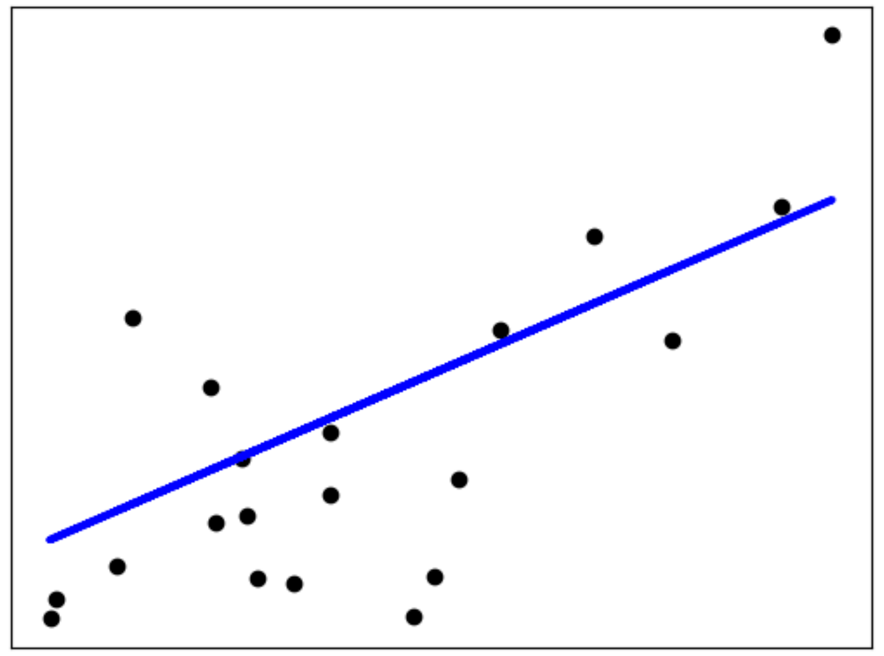
Machine learning models

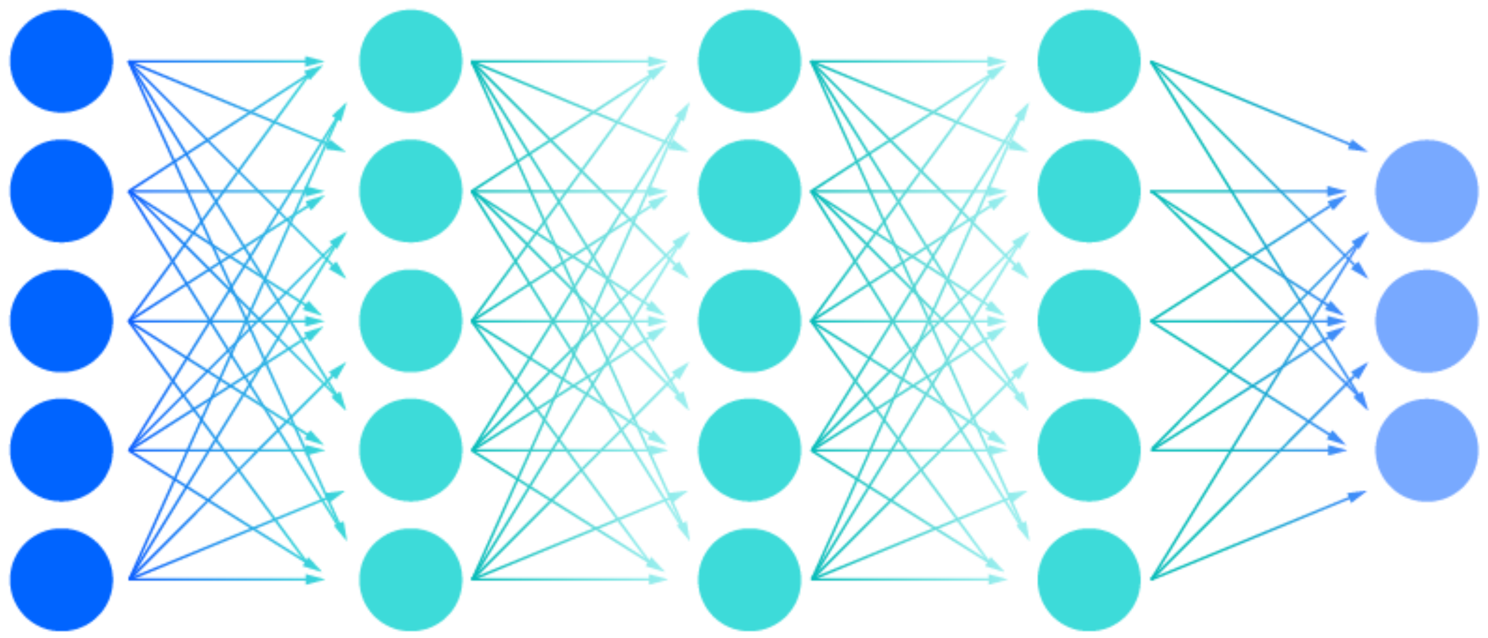
ML model: an algorithm used to perform a task.

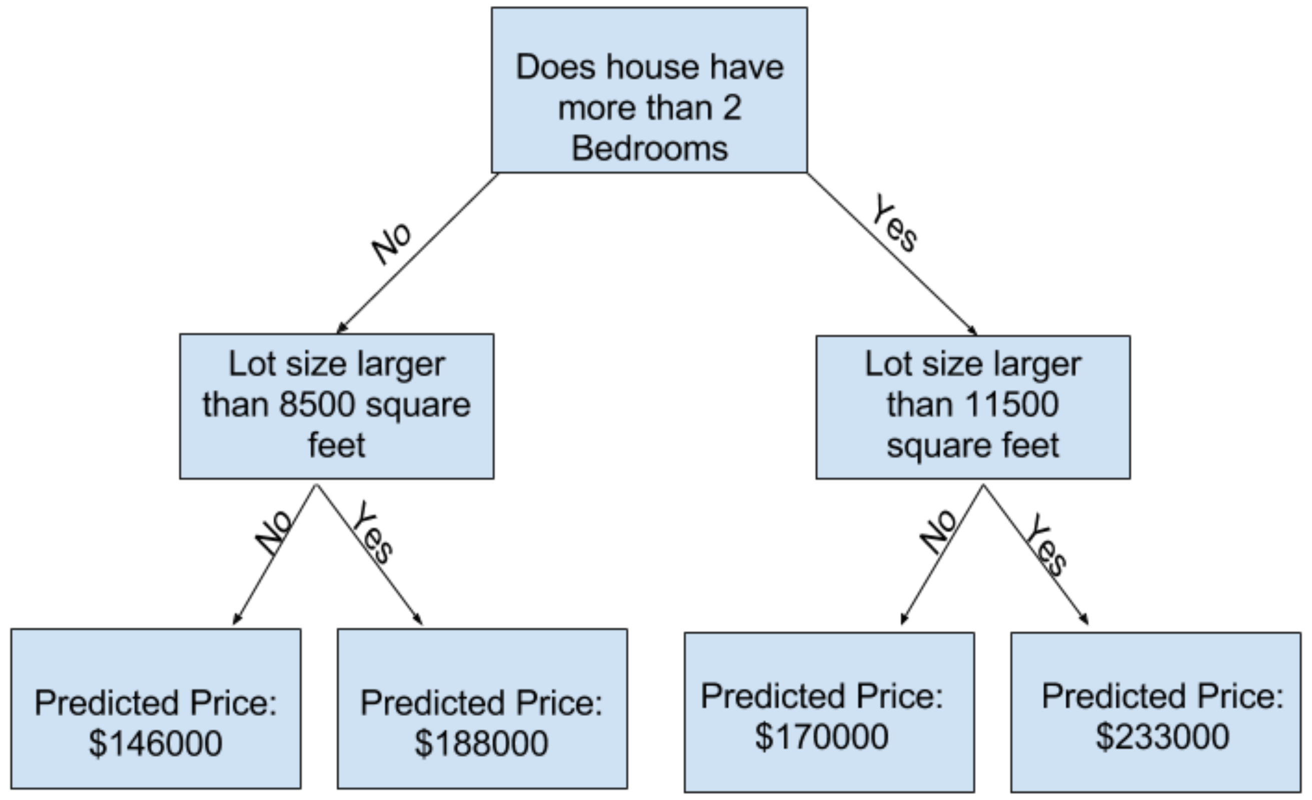
Supervised ML models:

- Linear models (e.g. perceptron)
- Deep neural networks*
  (e.g. multi-layer perceptron)
- Decision trees
- Ensembles (e.g. random forests)
- …

*also for unsupervised and
reinforcement learning







Credits

Lecture contents based on the following:

- COS 126 lecture on Introduction to Machine Learning by Ruth Fong
- COS 324 instructors and lectures
- Etienne Bernard, Wolfram. Introduction to Machine Learning: Machine Learning Paradigms. https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/
- Shree Nayar. Perceptron | Neural Networks. https://youtu.be/OFbnpY_k7js?si=wrWOvNXE3UVCq-7Z

Credits

Media	Source	License
AlphaFold	AlphaFold	CC BY 4.0
ChatGPT screenshot	Mashable	
Cicero	Science	
DALL-E 2	Open AI	
Dancing Robot	Disney Research	
Basketball Tabular Data	Statology	
Object detection NYC	Ilija Mihajlovic	
Chat GPT market	MarketWatch	
Robotic arms	Facebook Research	
Audio wave	COS 126	
Blood pressure	Wikimedia	CC0 1.0
Object segmentation	PASCAL VOC	
Segmentation for cars	pytorch-semseg	

Credits

Media	Source	License
Google Translate	Analytics Drift	
Clustering	Google Developers	
Dimensionality reduction	Wolfram	
PCA Colleges	Ruth Fong	
Anomaly detection	Wolfram	
StyleGAN	arXiv	
Reinforcement learning	Wolfram	
CartPole	Gymnasium	
Cart Pole Games	Andreas De Neve	
Atari Breakout	OpenAI Gym	
AlphaGo	New Yorker	
Rubik's Cube robot arm	OpenAI	
TossingBot	Google Research	

Credits

Media	Source	License
Laundry folding robot	NPR	
US election map	Wikimedia	CC0 1.0
Doctor lung scan	Pexels	
Golden retriever	Emojis.sh	
Doberman	Emojis.sh	
Neuron	Wikimedia	CC BY-SA 4.0
Linear regression	scikit-learn	
Deep neural network	IBM	
Decision tree	Kaggle	