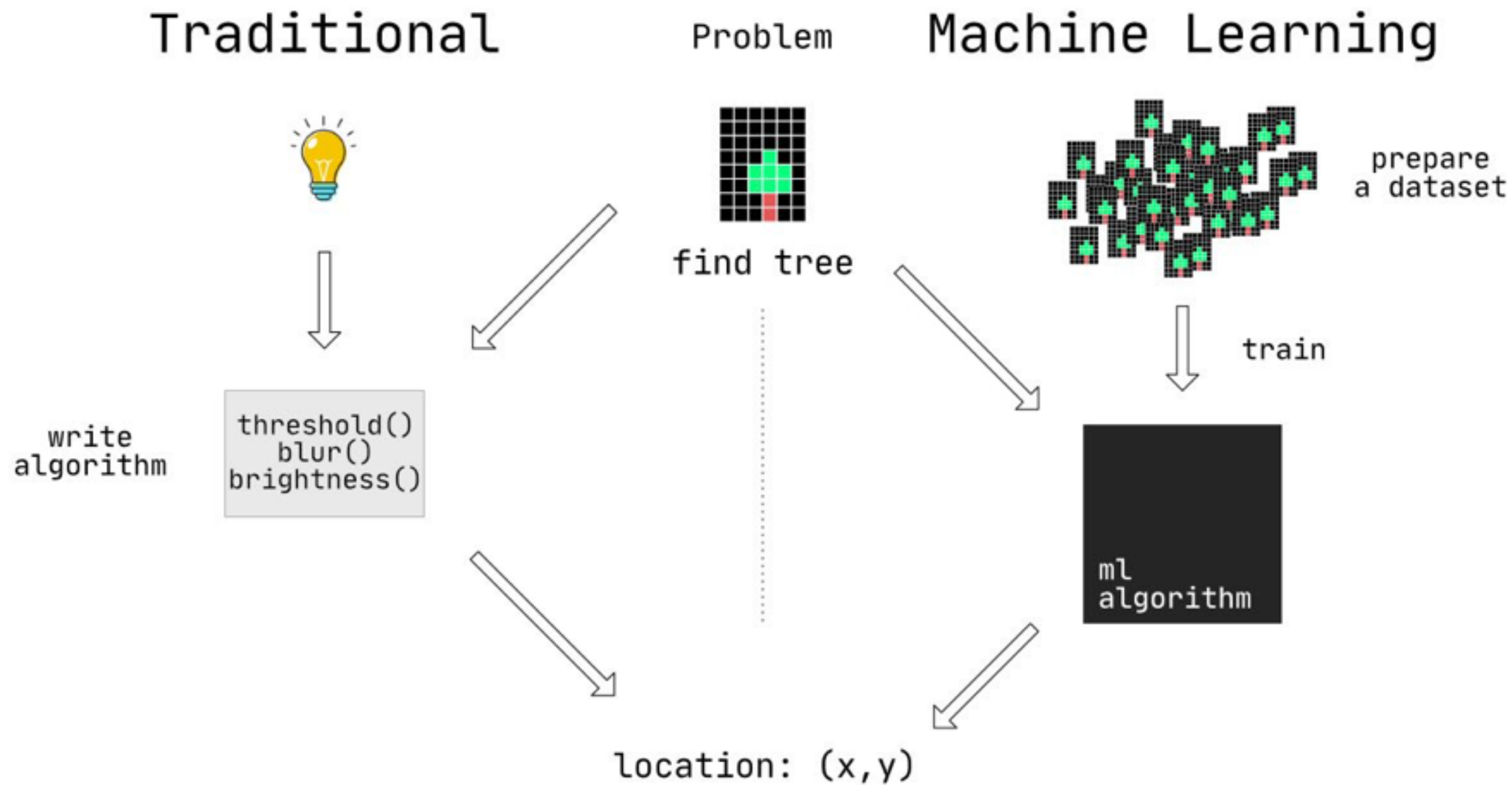


INTERACTION DESIGN

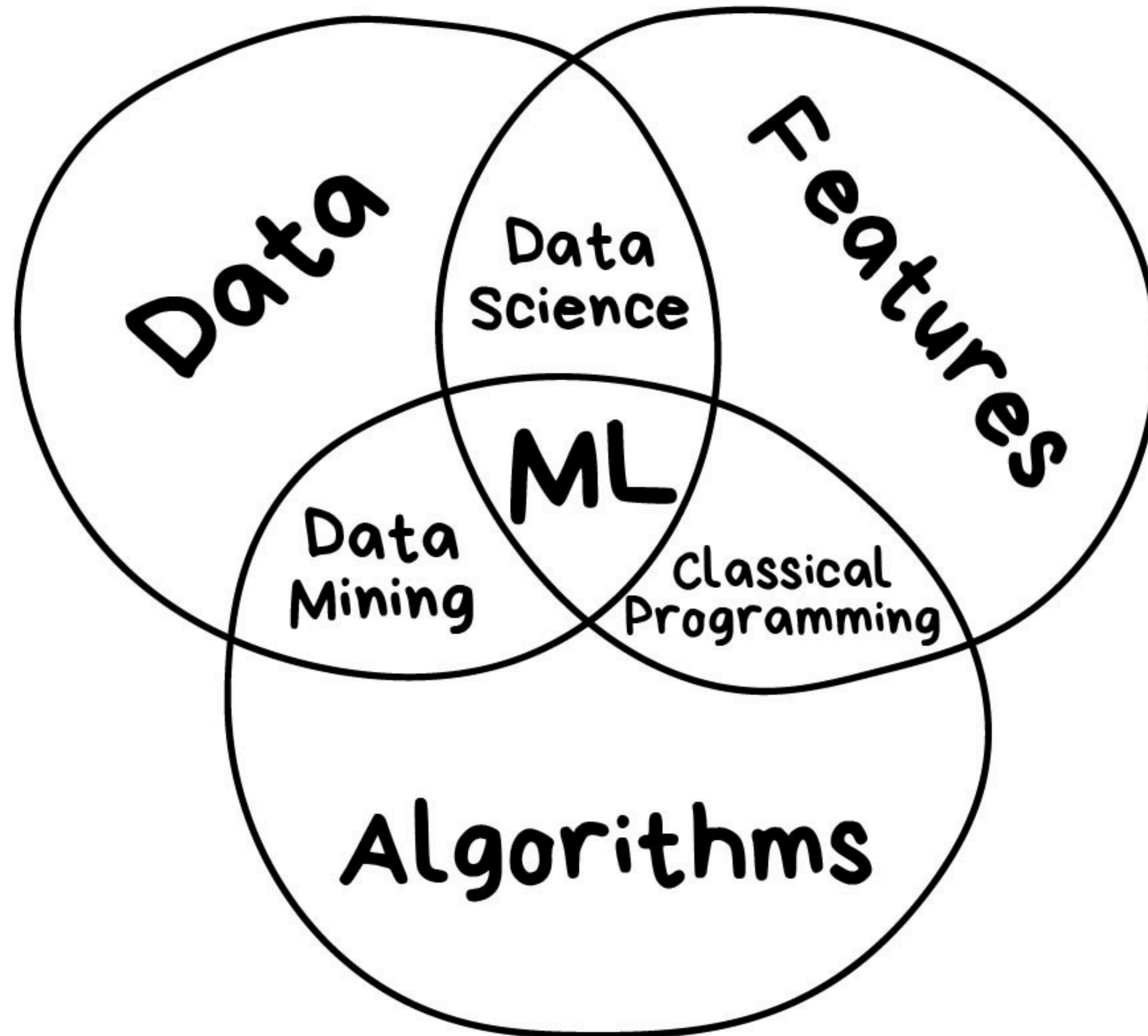
COMPUTER VISION & MACHINE LEARNING

Bits & Atoms IV

MACHINE LEARNING



MACHINE LEARNING

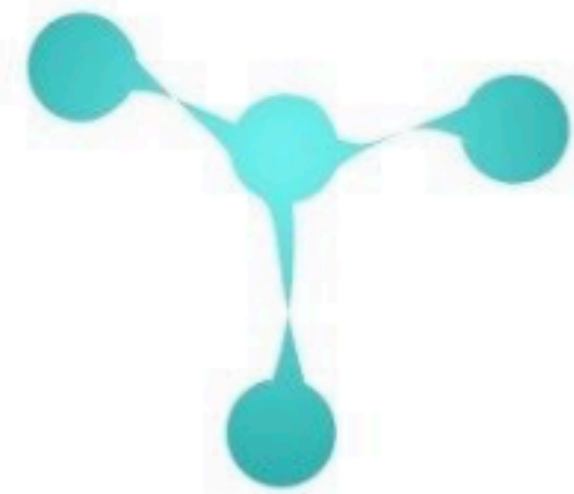


MACHINE LEARNING

 Microsoft
CNTK

Caffe

 **Caffe2**



PYTORCH


Chainer

 **K** **Keras**

 **TensorFlow**

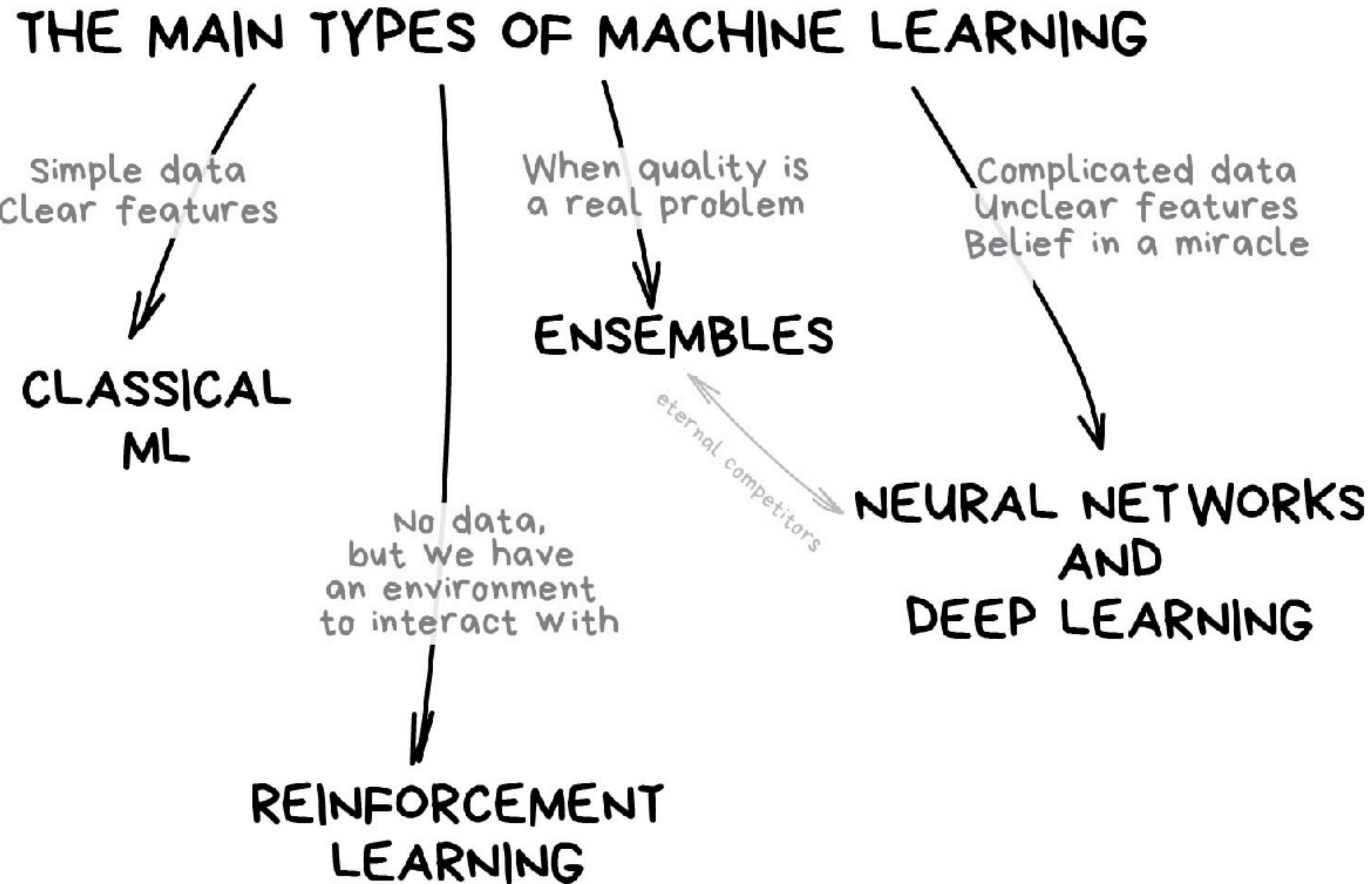
theano 

dy/net

 **mxnet**

 **GLUON**

MACHINE LEARNING



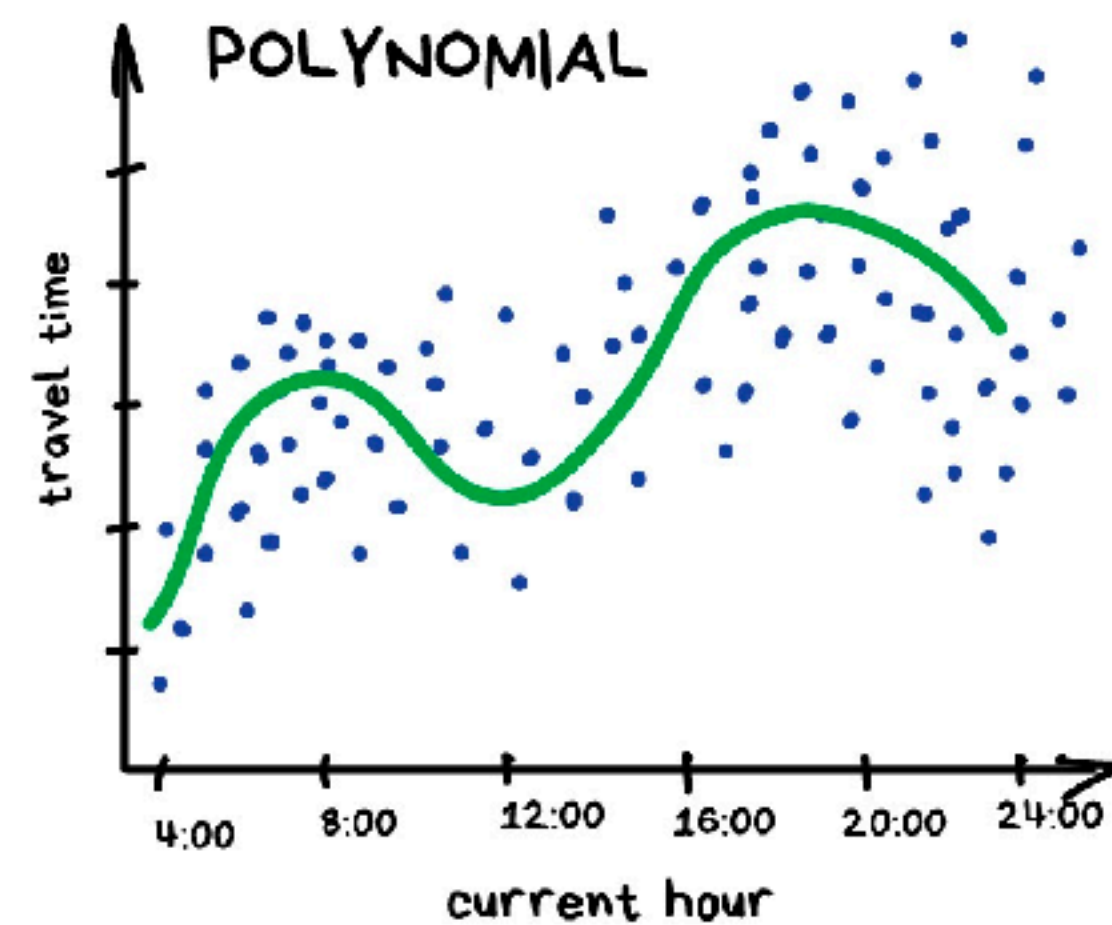
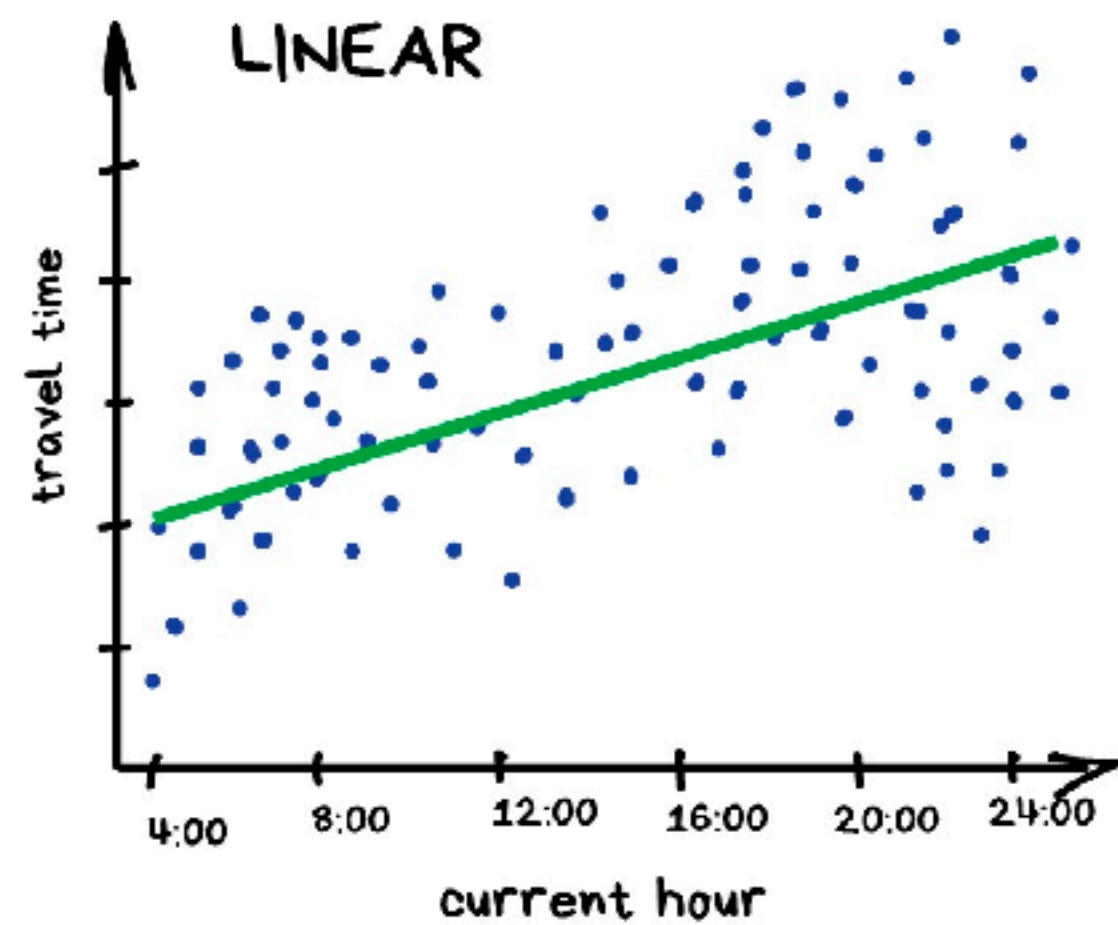
INTERACTION DESIGN

CLASSICAL MACHINE LEARNING

Bits & Atoms IV

REGRESSION

PREDICT TRAFFIC JAMS



REGRESSION

Use cases:

- Stock price forecasts
- Demand and sales volume analysis
- Medical diagnosis
- Number-time correlations

Popular algorithms:

- Linear
- Polynomial

REGRESSION

Regression predicts a **continuous target variable** Y . It allows you to estimate a value, such as housing prices or human lifespan, based on **input data** X .

$$Y = f(X) + \epsilon$$

X (input) = years of higher education

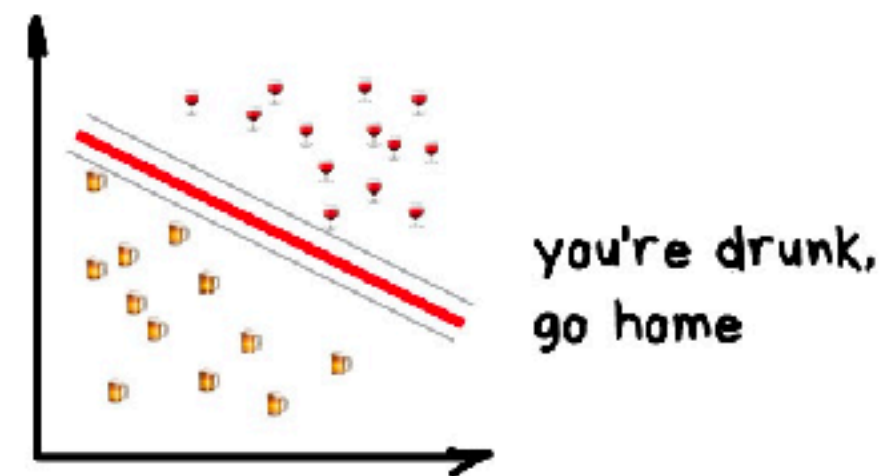
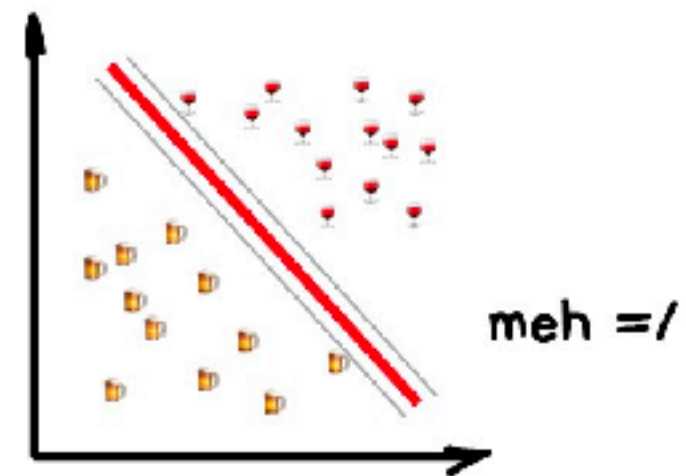
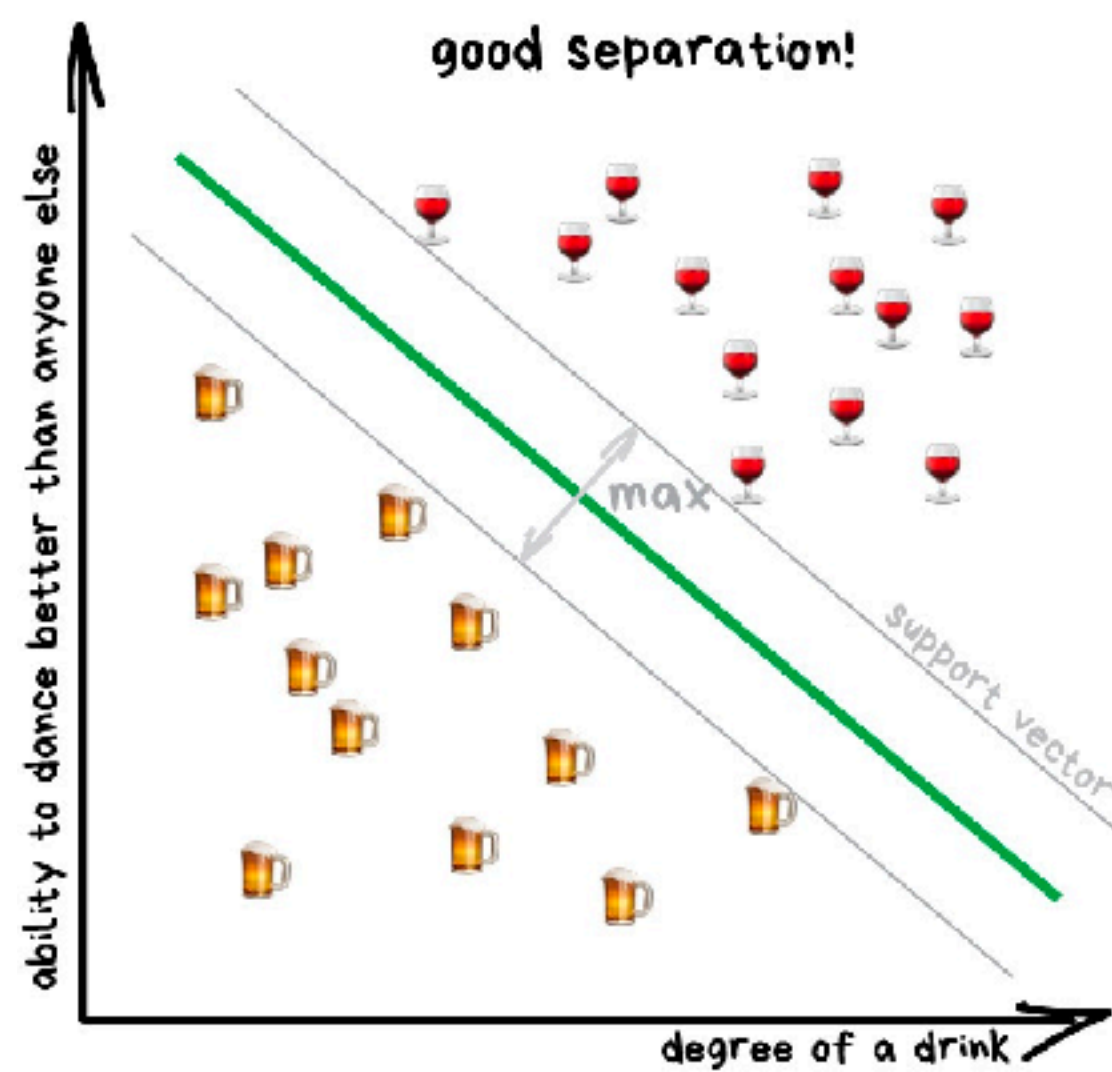
Y (output) = annual salary

f = function describing the relationship between X and Y

ϵ (epsilon) = random error term (positive or negative) with mean zero

CLASSIFICATION

SEPARATE TYPES OF ALCOHOL



Classification

Use cases:

- Spam filtering
- Language detection
- Search of similar documents
- Sentiment analysis
- Recognition of handwritten characters
- Fraud detection

Popular algorithms:

- Naive Bayes
- Decision Tree
- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machine

CLASSIFICATION

Classification predicts a discrete target label Y . It is the problem of assigning new observations to the class, based on a classification model built from labeled training data.

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x + \epsilon$$

What do you think are the **odds**($p/(1-p)$) that ZHdK will cancel Diploma exhibition again? Let's assume it's 4 times more likely to happen than not, making it **4-to-1 odds**.

$$P = 4/5$$

$$1-P = 1/5$$

$$P/(1-p) = 4.0$$

$$\ln(P/(1-p)) = 1.4$$

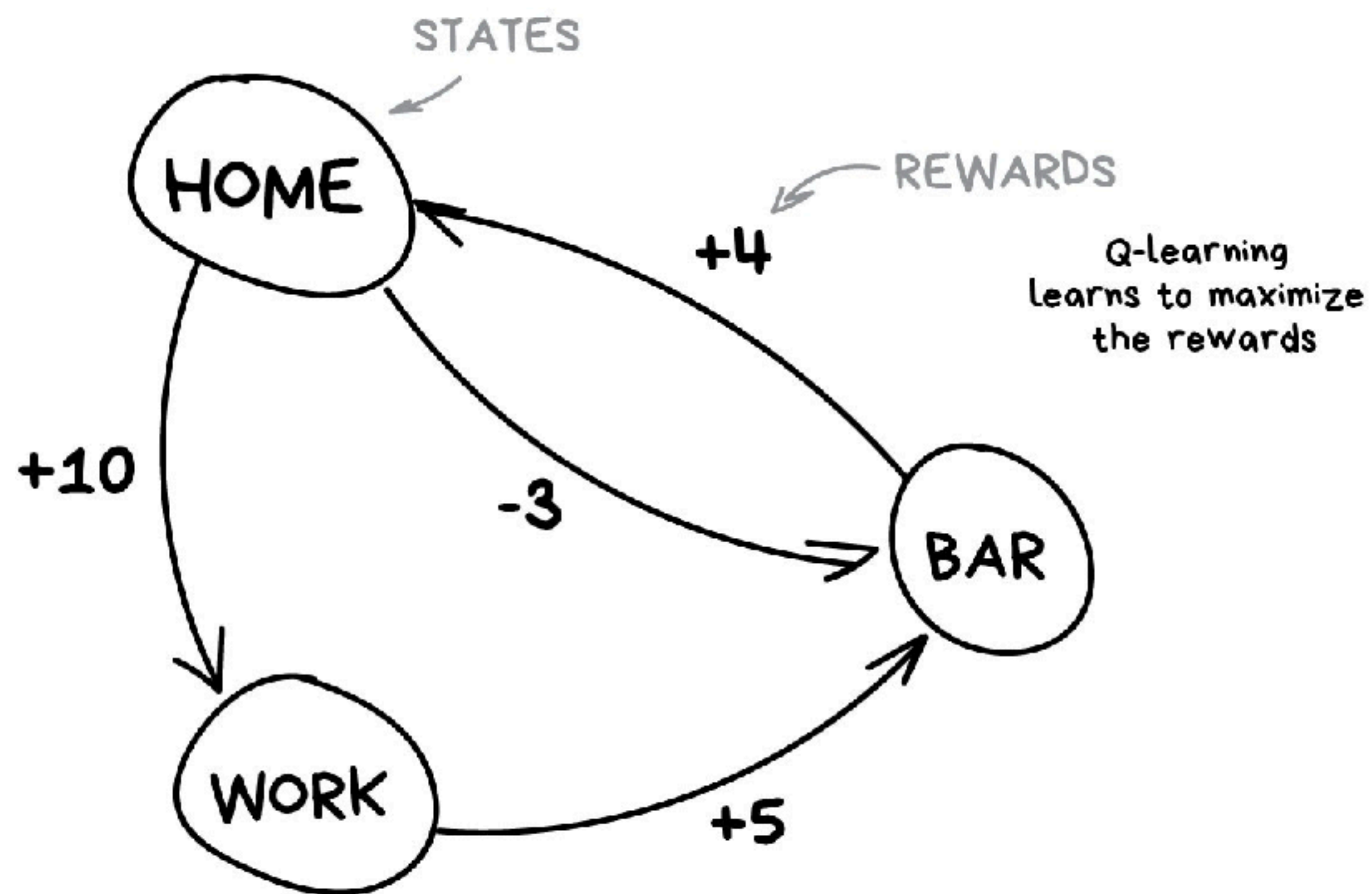
"It's gonna happen, the probability is 140 %"

INTERACTION DESIGN

REINFORCEMENT LEARNING

Bits & Atoms IV

REINFORCEMENT LEARNING



ROUTINE MARKOV PROCESS

Use cases:

- Self-driving cars
- Robot vacuums
- Games
- Automating trading
- Enterprise resource management

Popular algorithms:

- Q-Learning
- SARSA
- DQN,
- A3C
- Genetic algorithm

REINFORCEMENT LEARNING

In reinforcement learning there's no answer key, but your reinforcement learning agent still has to decide how to act to perform its task. In the absence of existing training data, the machine through trial-and-error it attempts its task, with the goal of maximizing long-term reward.



Crystal Maiden's Freezing Field
and **Viper's Nethertoxin** force
the **human team** to scatter



dotaarena-1
24

BREAK
25

STUNNED
22

dire2 (Bot)
23
+120

+285

dire4 (Bot)
24

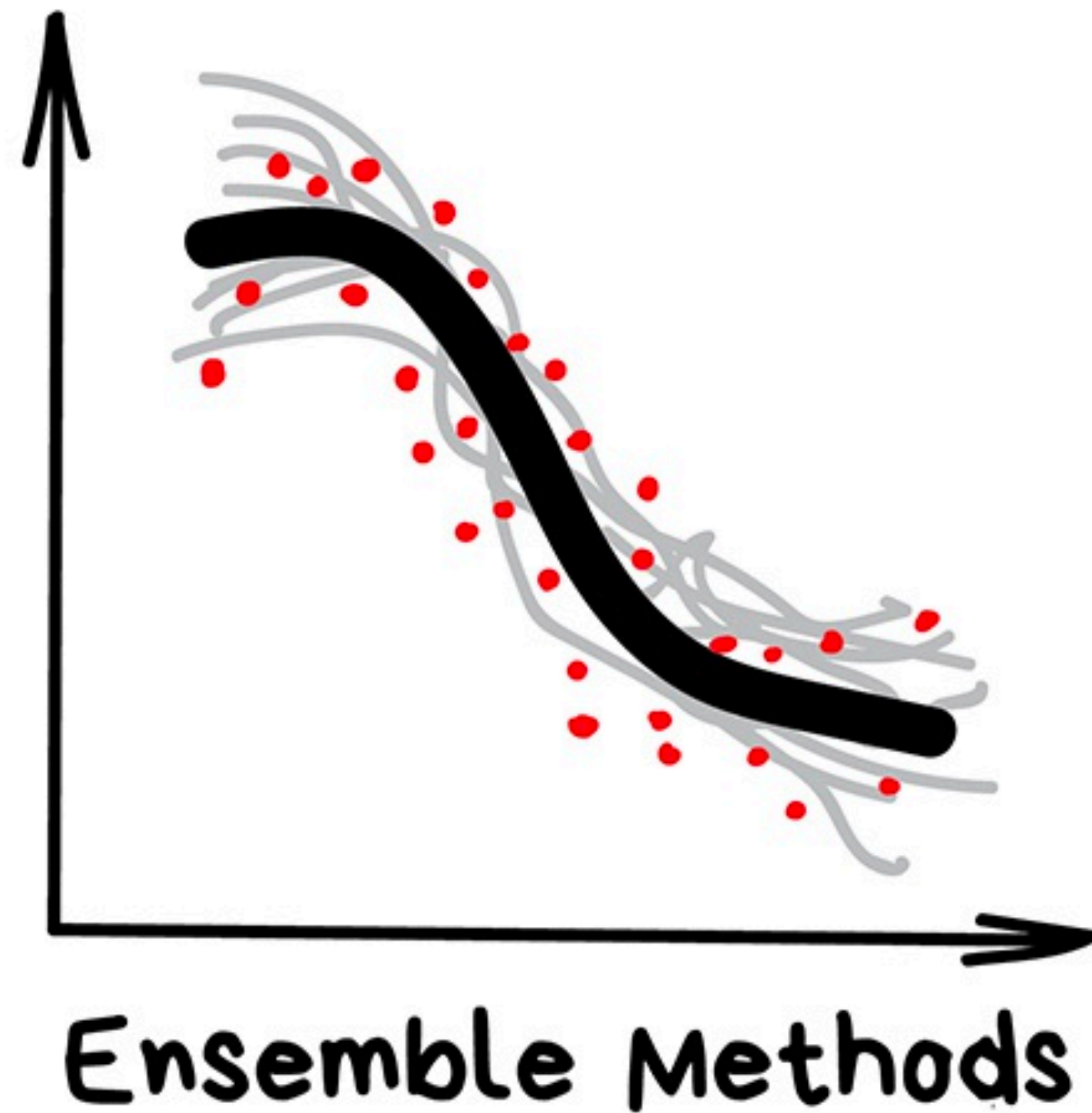
STUNNED
24

INTERACTION DESIGN

ENSEMBLE METHODS

Bits & Atoms IV

ENSEMBLE METHODS



Use cases:

- Classical algorithm approaches
- Search systems
- Computer vision
- Object detection

Popular algorithms:

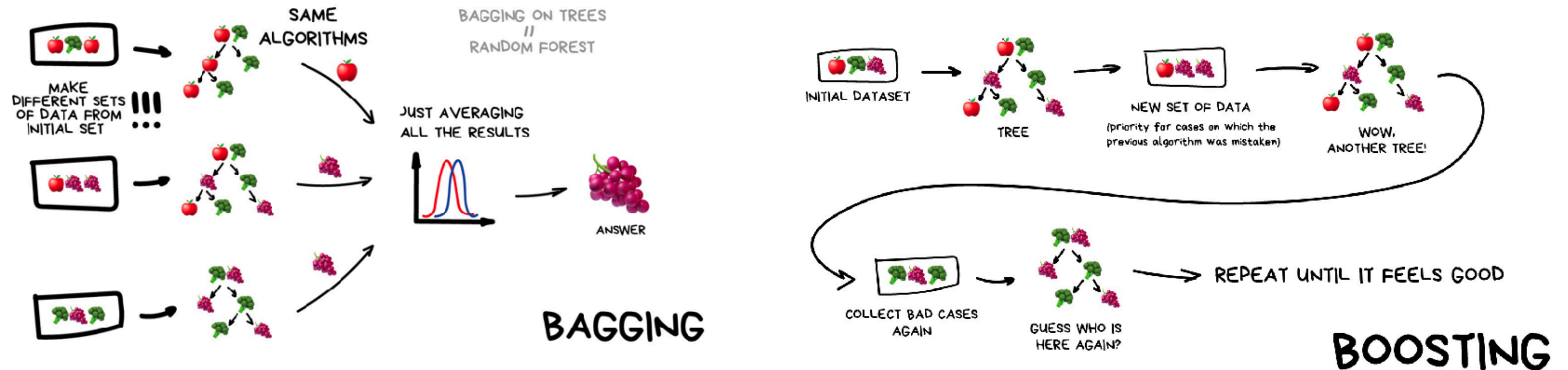
- Random Forest
- Gradient boosting

ENSEMBLE METHODS

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model.

e.x regression + decision tree + support vector machine + ...

Extremely high precision of classification !



INTERACTION DESIGN

NEURAL NETWORKS

Bits & Atoms IV

COMPUTER VISION VS DEEP LEARNING

COMPUTER VISION

Computer vision is a subset of machine learning that deals with making computers or machines understand human actions and languages

The goal is to interpret visual information in order to make better sense of the digital data.

Defect detection, image labelling, face recognition, object detection, image classification, object tracking, movement analysis, cell classification

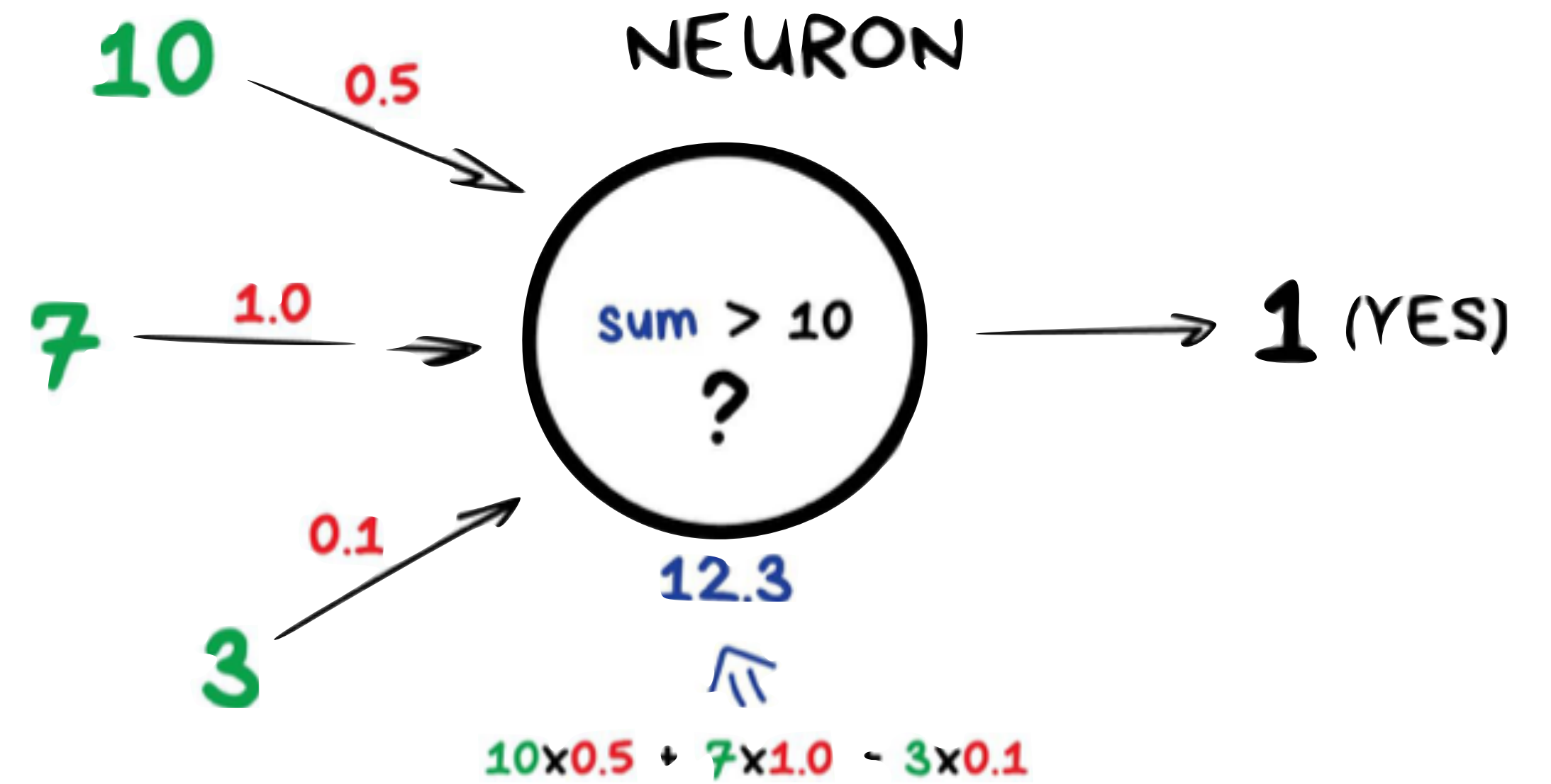
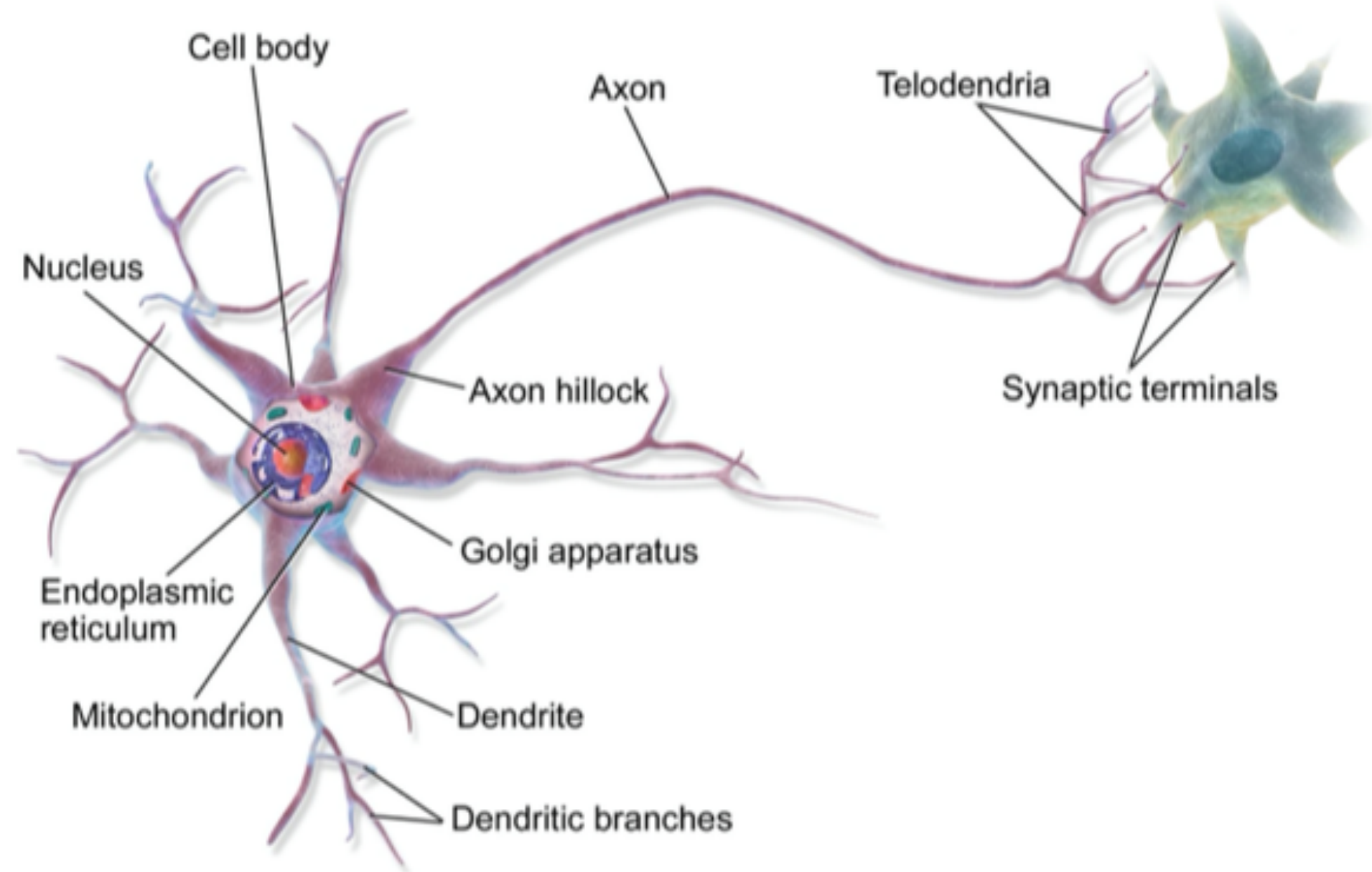
DEEP LEARNING

Deep learning is a subset of AI that seeks to mimic the functioning of the human brain based on artificial neural networks.

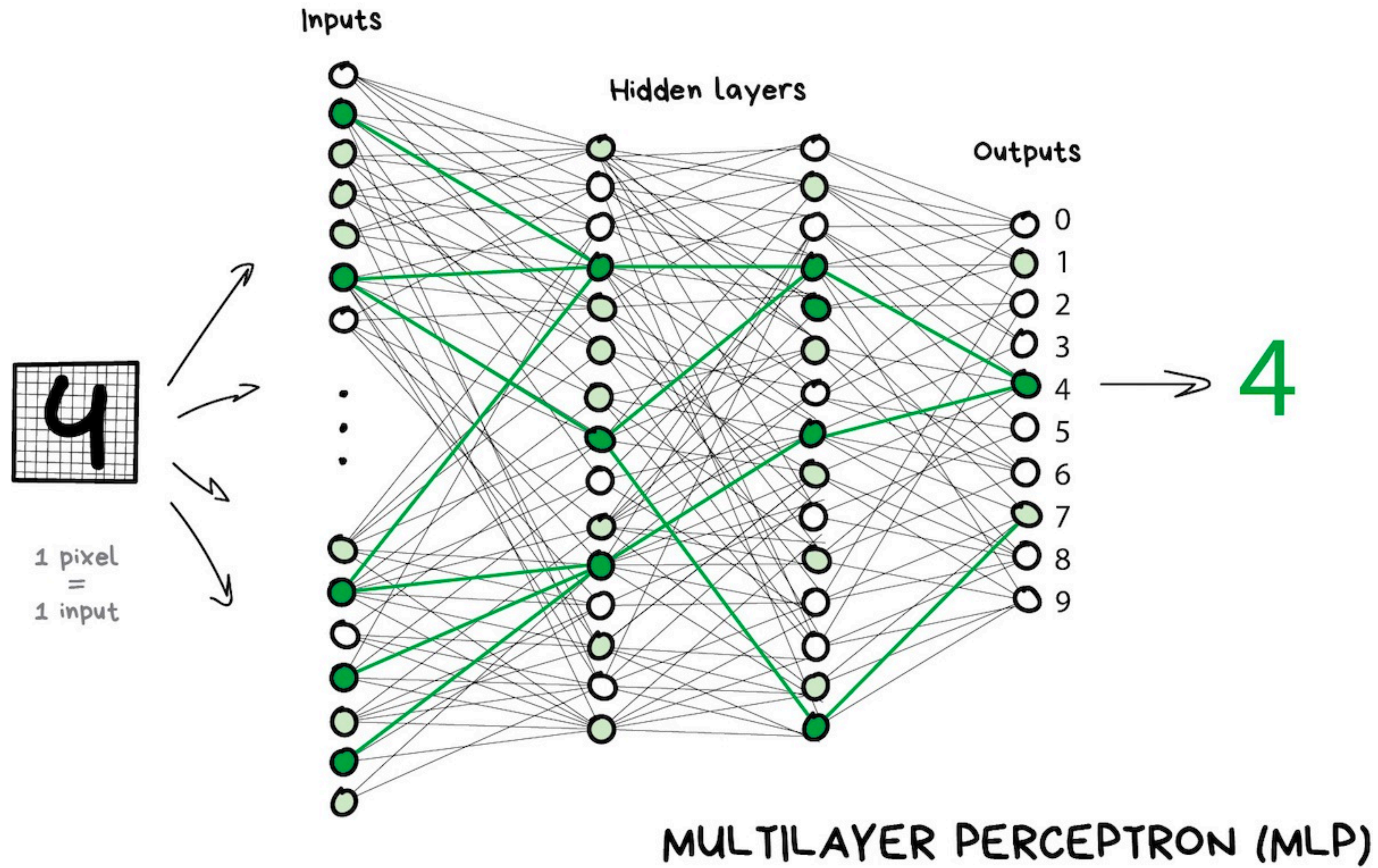
The goal is to extract features from raw data based on the notion of artificial neural networks.

Driving cars, natural language processing, visual recognition, image and speech recognition, virtual assistants, chatbots, fraud detection

NEURAL NETWORKS



NEURAL NETWORKS

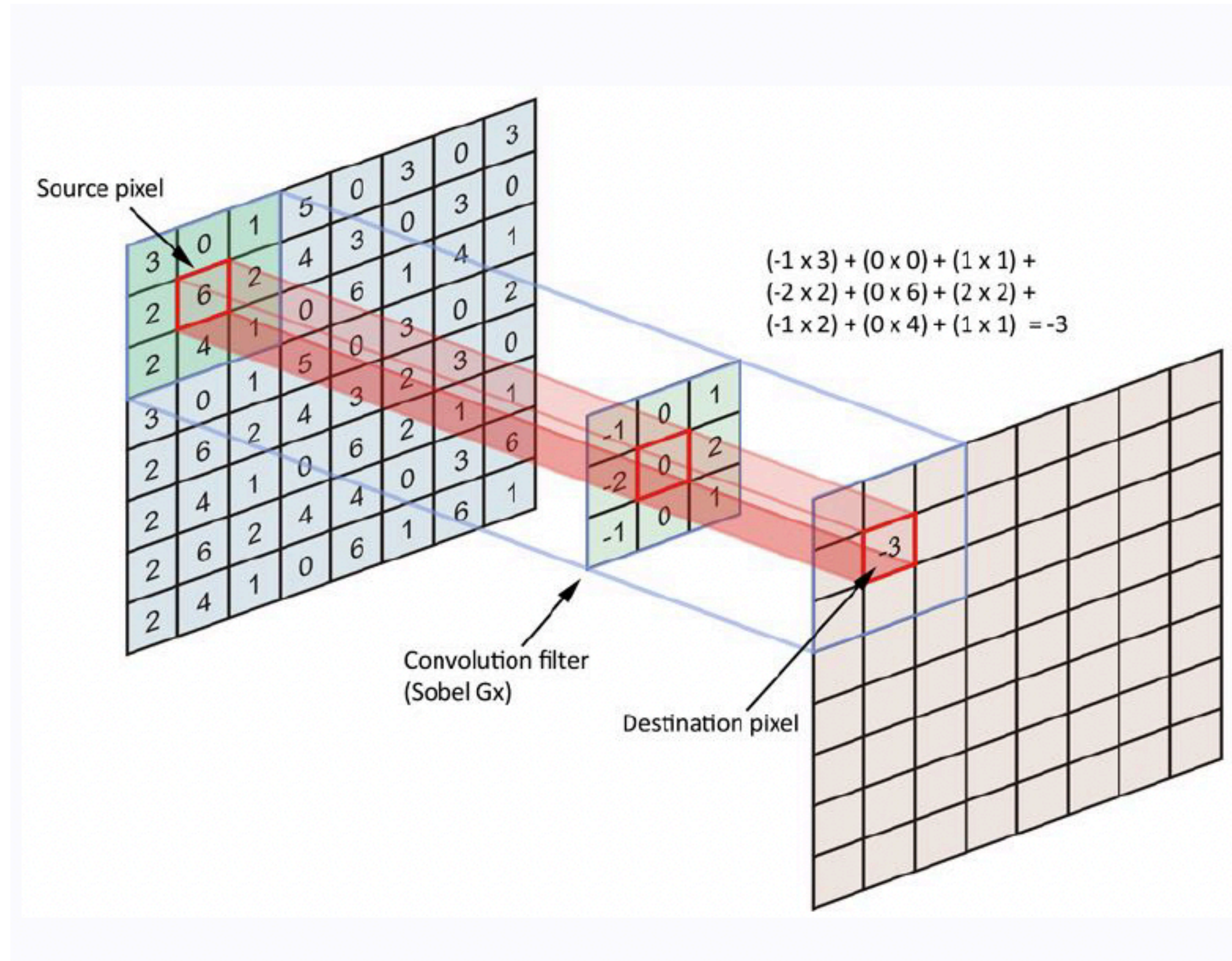


INTERACTION DESIGN

CONVOLUTIONAL NEURAL NETWORKS

Bits & Atoms IV

CNN



CNN are used for images and audio

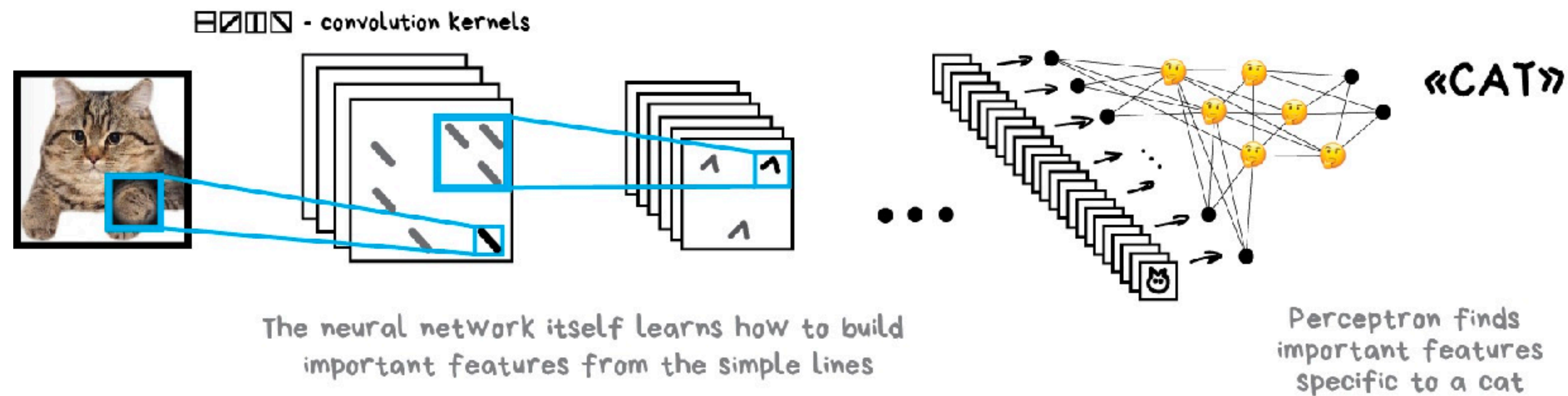
Use cases:

- Search for objects on photos
- Face recognition
- Style transfer
- Object detection
- Enhancement
- Slow-mo/Speedup effects

CNN

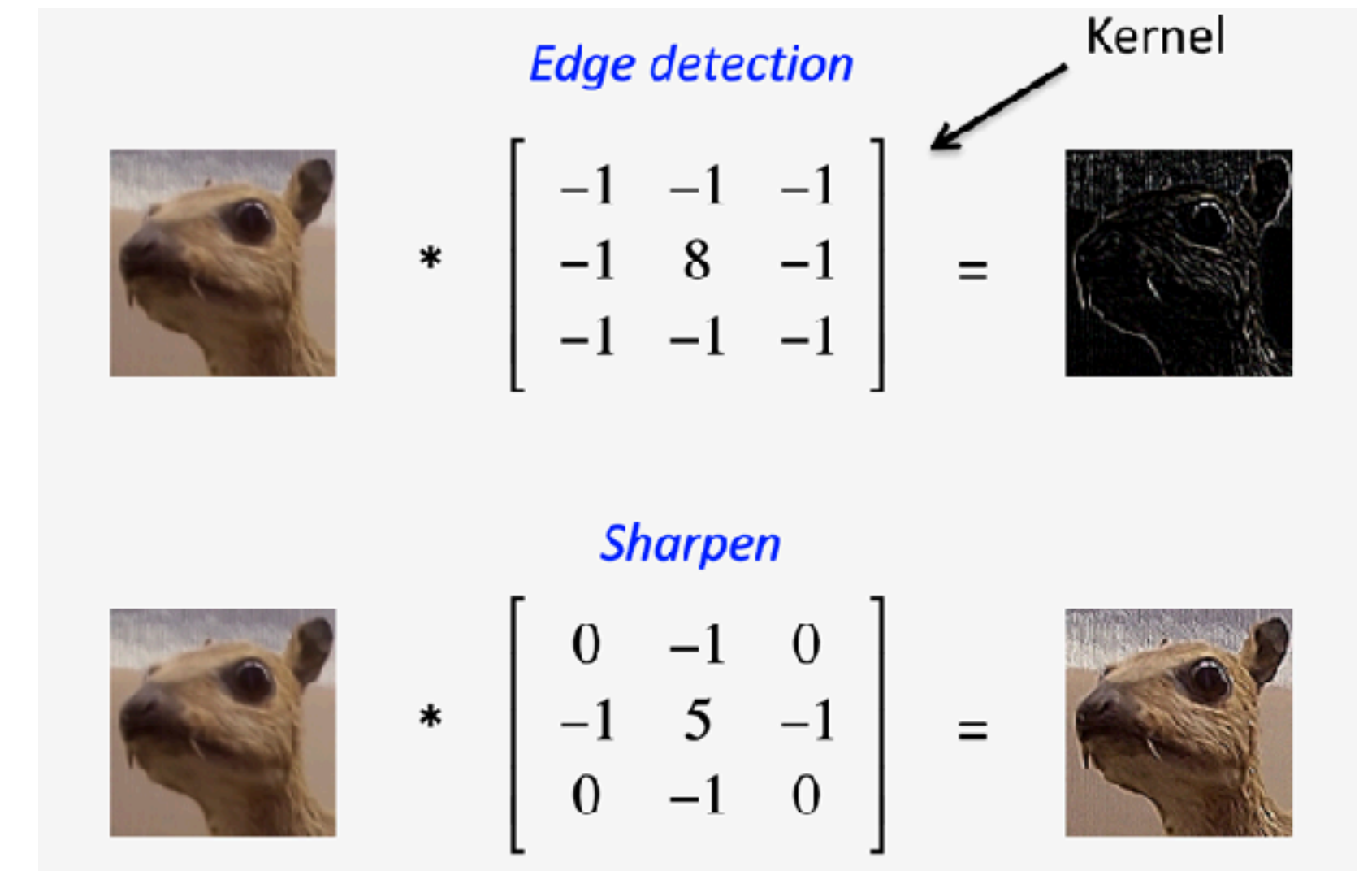
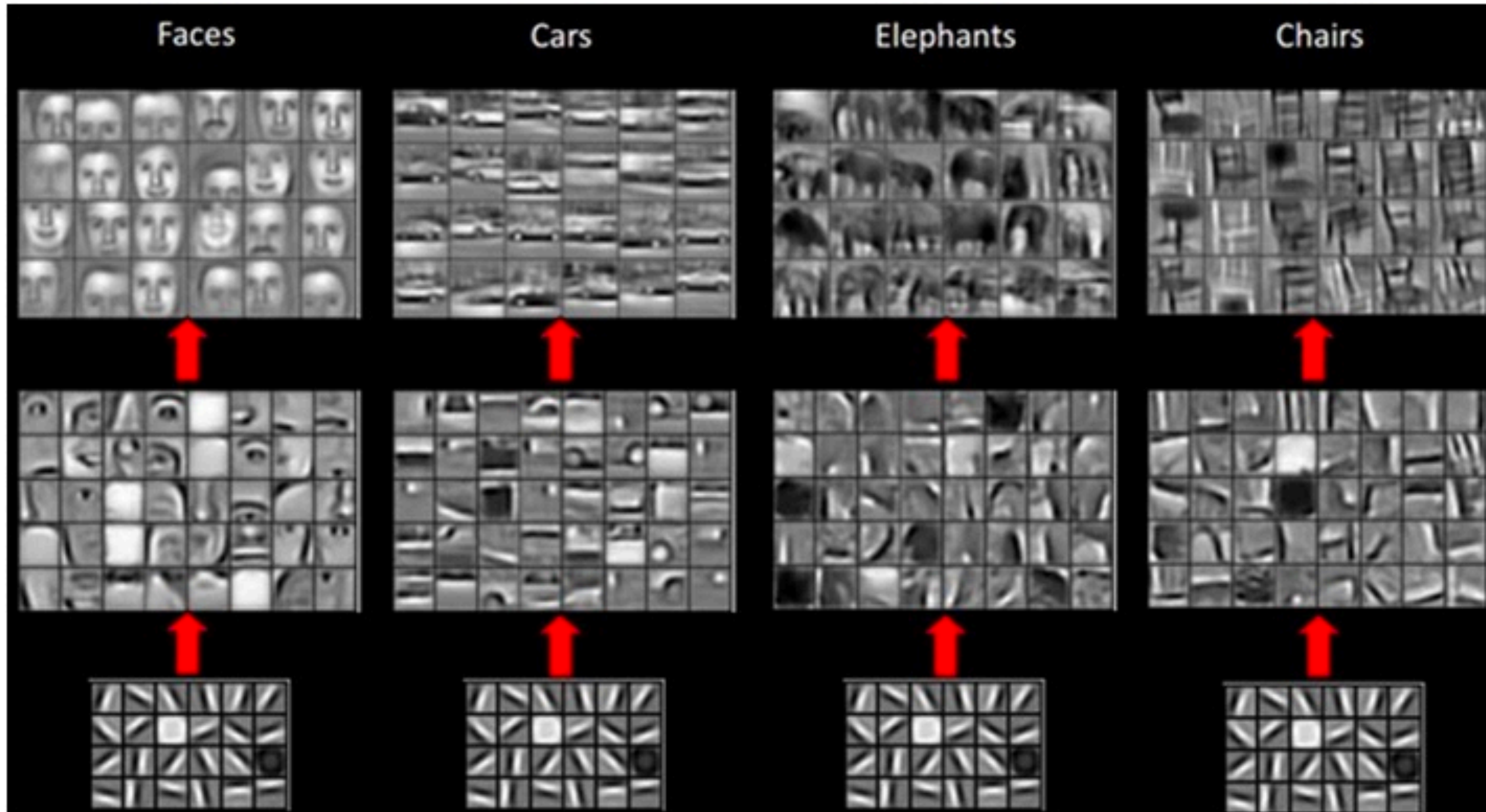
The machine learns features, building its own interpretation of an image on top of basic lines e.x. an algorithm first divides the whole image into 8×8 pixel blocks and assigns to each a type of dominant line - either horizontal [-], vertical [|] or diagonal [/] representing objects edges on the image.

This operation is called **convolution**.



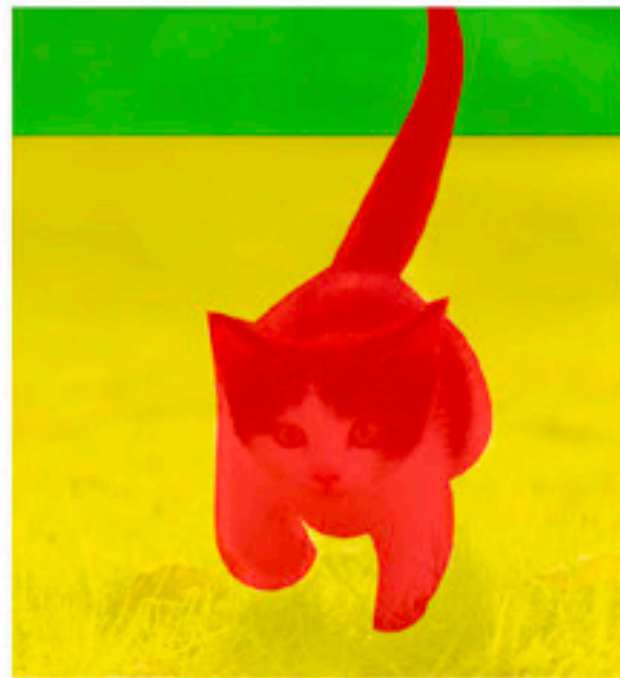
CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN



CNN

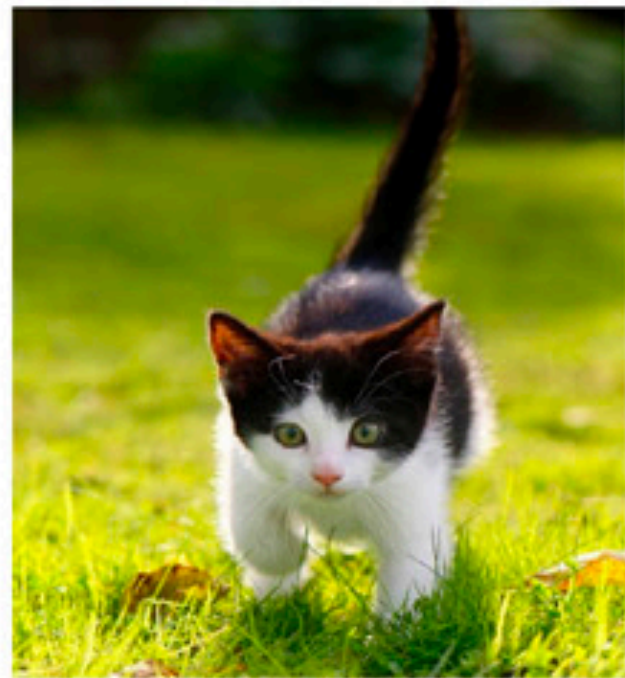
Semantic Segmentation



**CAT GRASS
TREE**

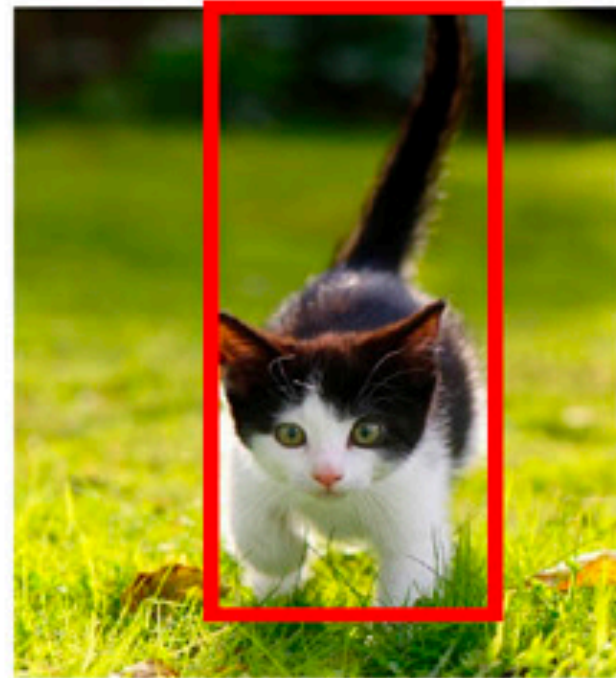
No object
Just pixels

Classification



CAT

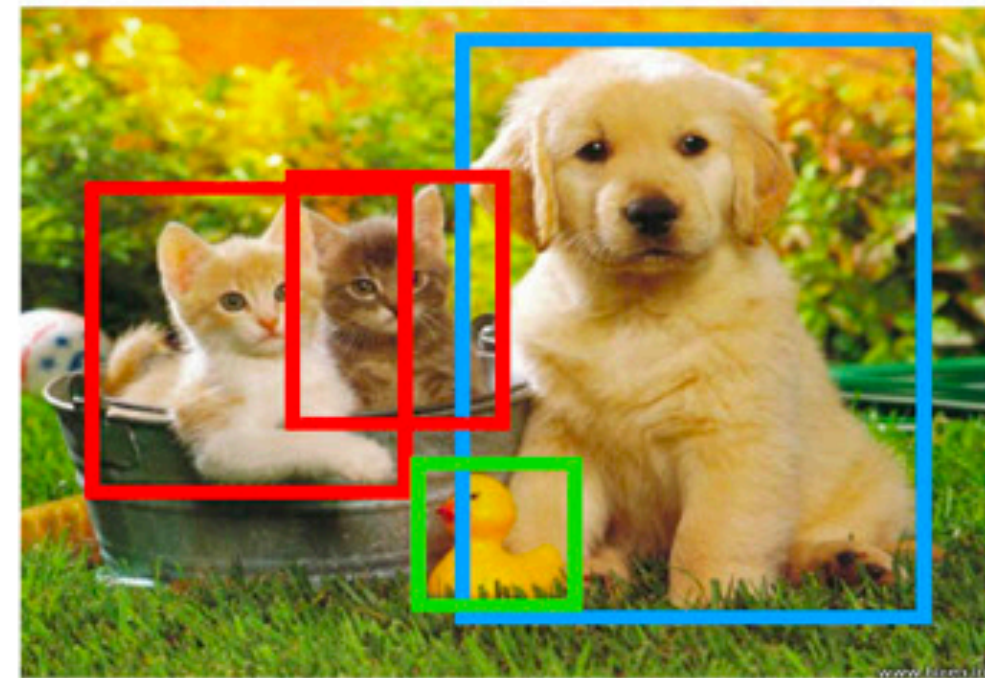
Classification + localization



CAT

Single object

Object detection



CAT DOG DUCK

Multiple objects

Instance segmentation



CAT CAT DOG DUCK

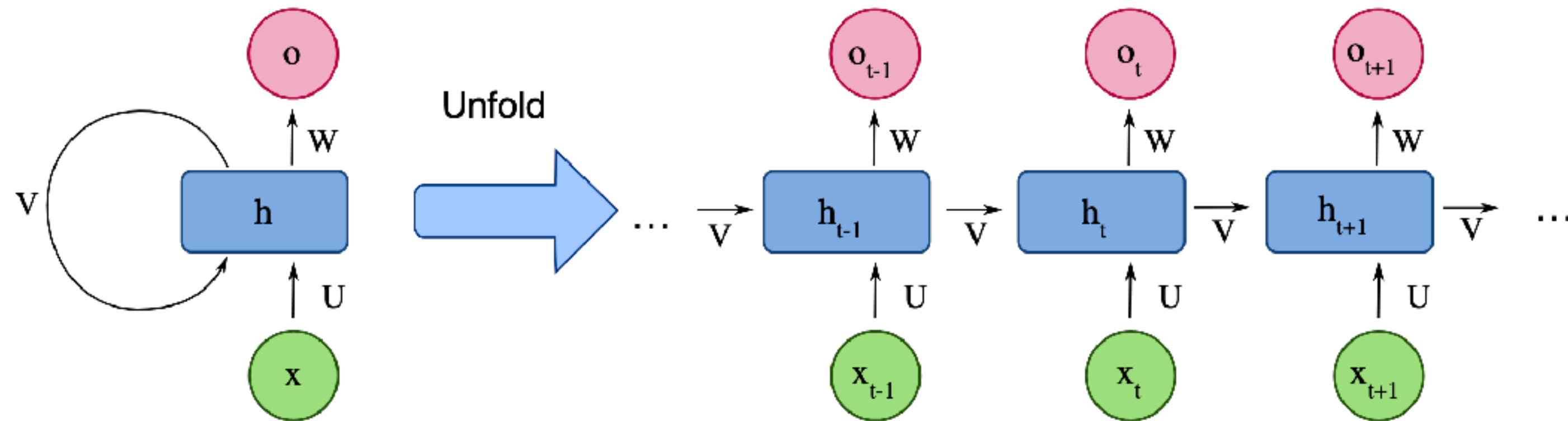


INTERACTION DESIGN

RECURRENT NEURAL NETWORKS

Bits & Atoms IV

RNN



RNN are used for speech and soundwaves.

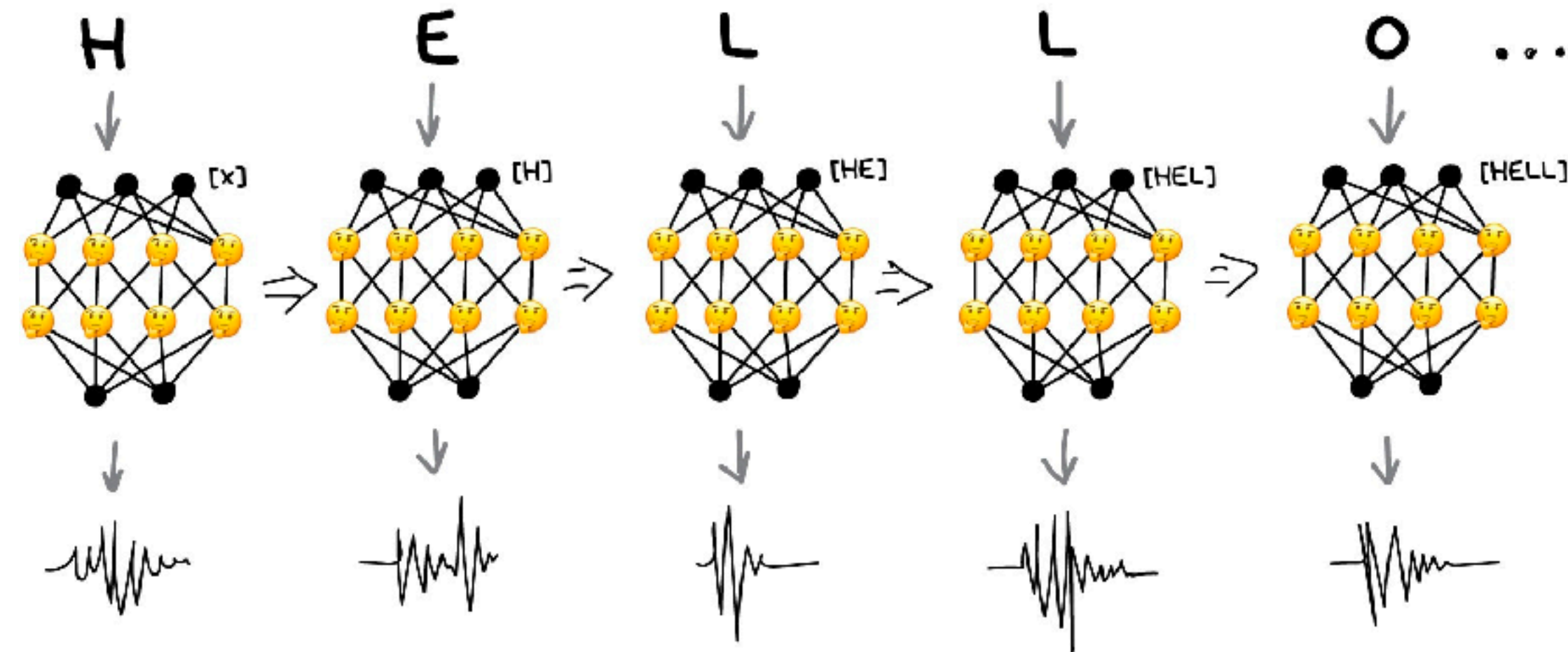
Use cases:

- Machine translation
- Voice synthesis
- Natural Language Processing
- Speech recognition
- Image captioning

RNN

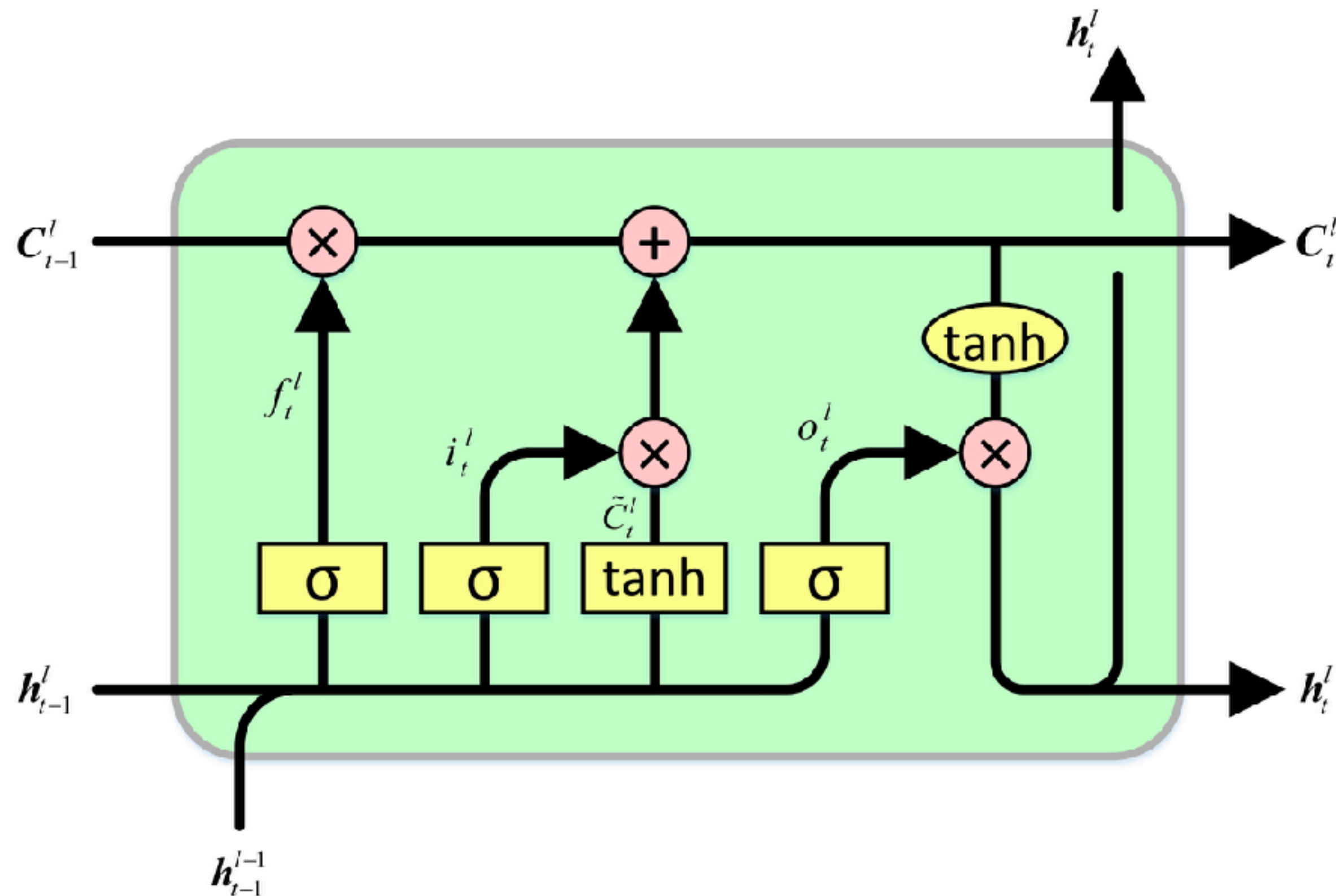
A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor

They might be able to connect previous information to the present task, such as using previous letter might inform the understanding of the pronunciation of the next letter. This process is **recurring**.



RECURRENT NEURAL NETWORK (RNN)

RNN



Essential to RNN the use of **Long Short Term Memory networks (LSTMs)** a very special kind of recurrent neural network which are capable of learning both short and long-term dependencies.

LSTMs also have this chain like structure, but the repeating module has a different structure with in-built RAM-like memory model.



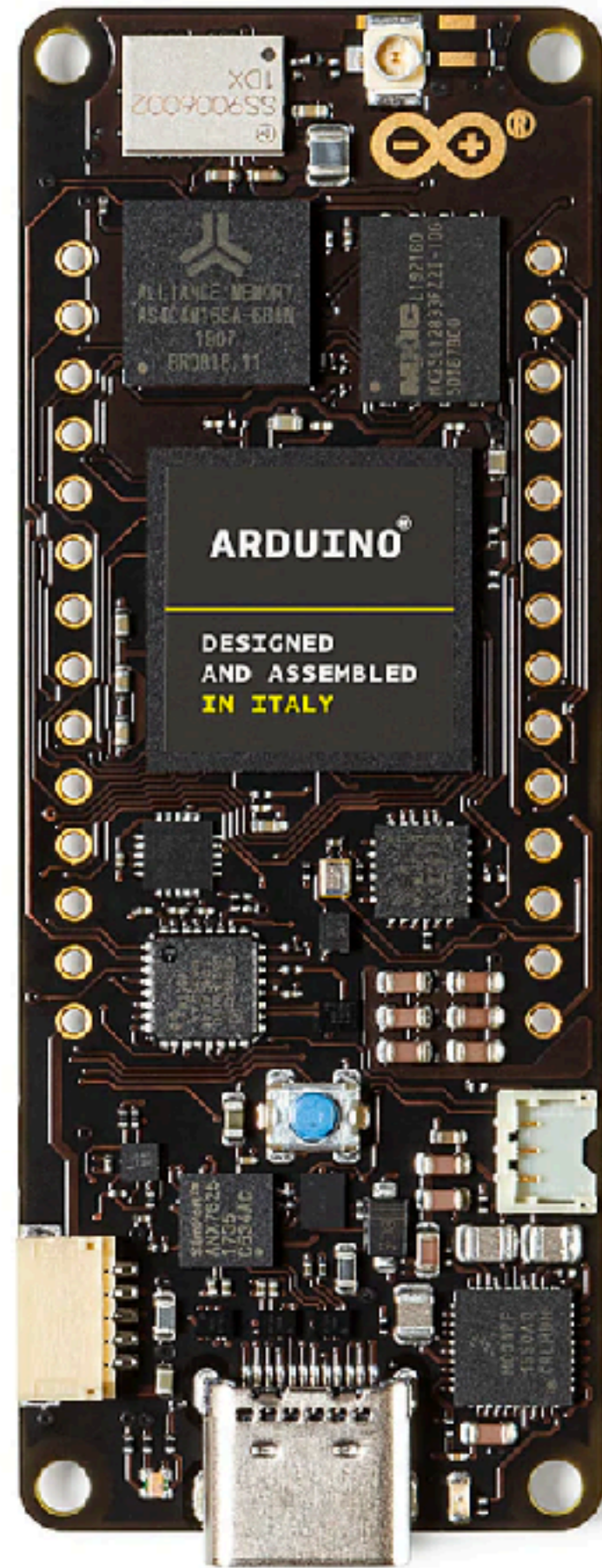
Video source: Team Coco, <https://www.youtube.com/watch?v=UT7h4nRcWjU>

INTERACTION DESIGN

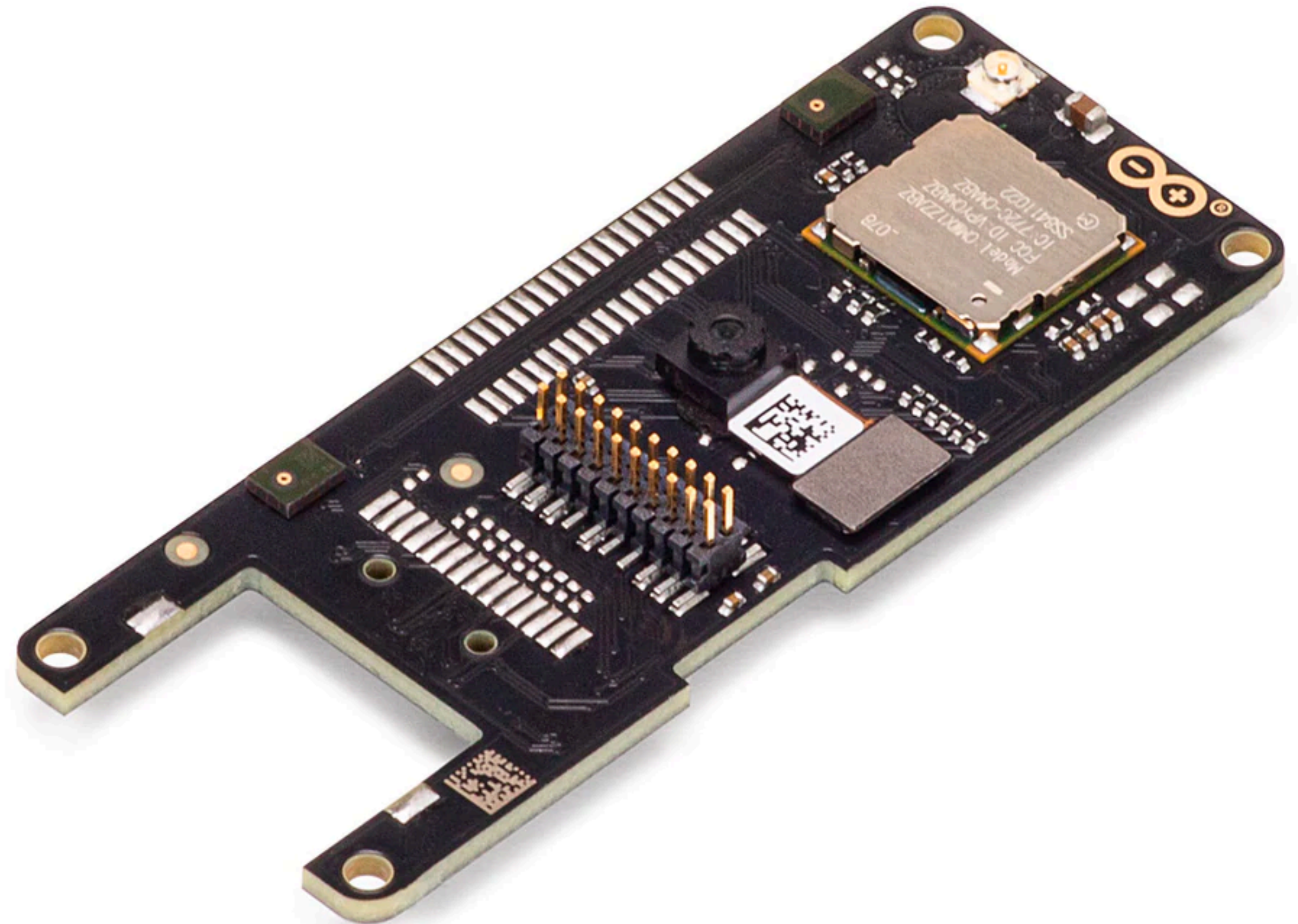
LORA + MACHINE LEARNING

Bits & Atoms IV

LORA + COMPUTER VISION



Arduino Portenta H7



Portenta Vision Shield

LORA + COMPUTER VISION

The screenshot displays the OpenMV IDE interface. The main window shows a Python script named `ei_image_classification.py` with the following code:

```
1 # Edge Impulse - OpenMV Image Classification Example
2
3 import sensor, image, time, os, tf
4
5 sensor.reset() # Reset and initialize the sensor.
6 sensor.set_pixformat(sensor.GRAYSCALE) # Set pixel format to RGB565 (or GRAYSCALE)
7 sensor.set_framesize(sensor.QVGA) # Set frame size to QVGA (320x240)
8 sensor.set_windowing((240, 240)) # Set 240x240 window.
9 sensor.skip_frames(time=2000) # Let the camera adjust.
10
11 net = "trained.tflite"
12 labels = [line.rstrip('\n') for line in open("labels.txt")]
13
14 clock = time.clock()
15 while(True):
16     clock.tick()
17
18     img = sensor.snapshot()
19
20     # default settings just do one detection... change them to search the image...
21     for obj in tf.classify(net, img, min scale=1.0, scale mul=0.8, x_overlap=0.5, y_overlap=0.5):
22         print("*****\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
23         img.draw_rectangle(obj.rect())
24         # This combines the labels and confidence values into a list of tuples
25         predictions_list = list(zip(labels, obj.output()))
26
27         for i in range(len(predictions_list)):
28             print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
29
30     print(clock.fps(), "fps")
31
```

The right side of the interface shows a live camera feed in the "Frame Buffer" window. Below it, the "Histogram" window displays the RGB color space for the image. The histogram shows three distributions: Red (R), Green (G), and Blue (B). The statistics for each channel are as follows:

Channel	Mean	Median	Mode	StDev	Min	Max	LQ	UQ
R	70	66	66	27	16	255	58	74
G	70	65	69	27	16	255	61	73
B	70	66	66	27	16	255	58	74

The bottom status bar shows the following information: Board: H7, Sensor: HM01B0, Firmware Version: 3.9.3 - [latest], Serial Port: ttyACM0, Drive: /media/bigcatbrother/2821-0049, FPS: 3,4.

Using TinyML + OpenMV

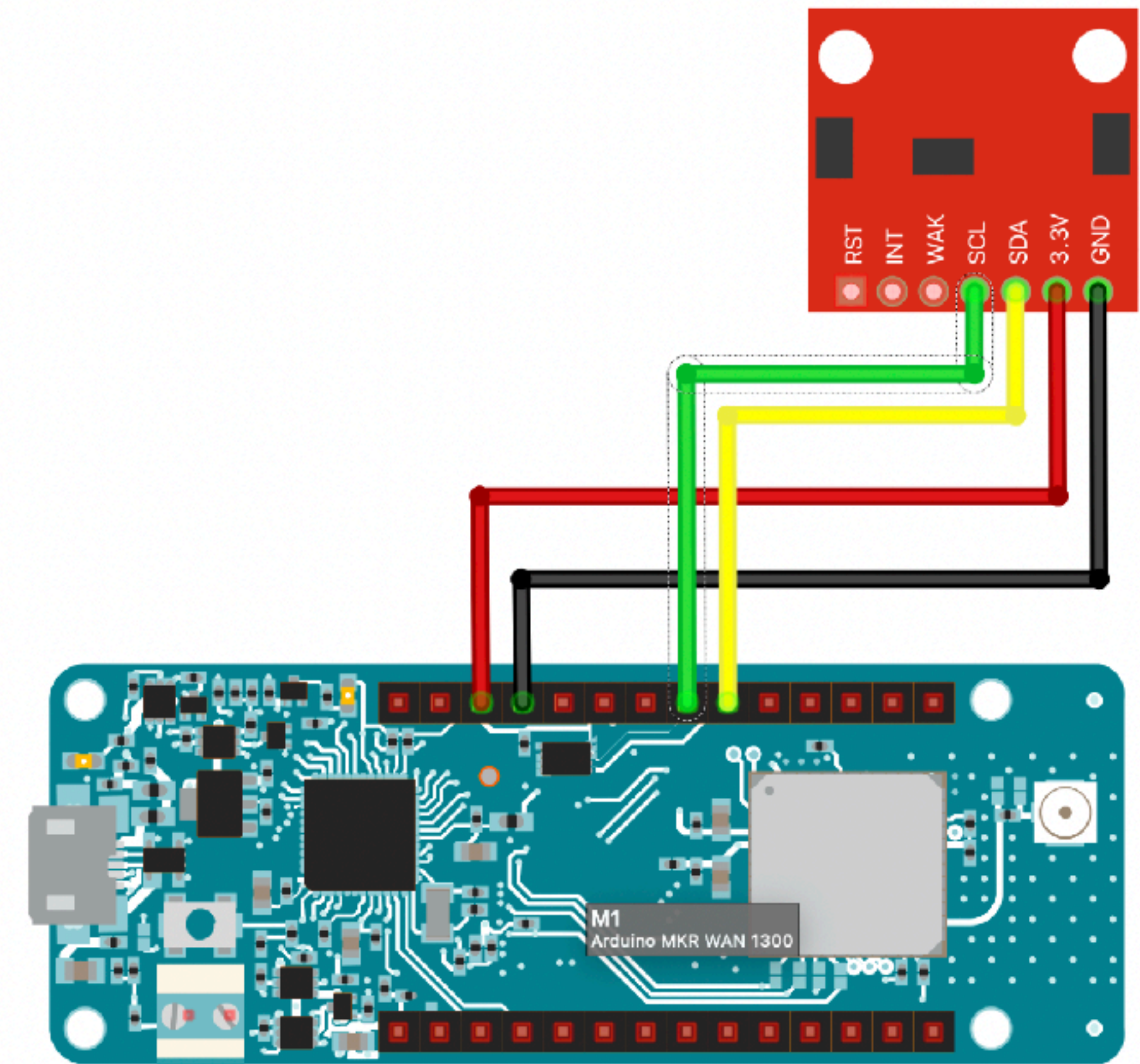
GROUP EXERCISE

CCS811 Air Quality Sensor :

- TVOCs in PPB
- equivalent CO2 in PPM

BME280 Temperature Humidity Pressure Sensor:

- Temperature
- Humidity
- Barometric Pressure



GROUP EXERCISE

Divide yourself in four groups; two Receivers(**R**) and two Senders(**S**).

S: Hookup environmental sensor to MKR 1310 in the same way as last week and open up the sketch which sends the data.

R: Open up the sketch which receives the data.

R and S: Choose together one or two sensor values and collect data from two different environments. Make sure you end up with one training data set (1000 examples) and one test data set (100 examples)

R: Train ML model which either:

- Recognises or predicts the environment for the new values.
- Predicts one sensor value based on the other

Use the provided templates to create your own model.

R + S: Create a sketch that represents the trained model in a visual way.

GROUP EXERCISE

Feel free to explore other libraries

<https://javascriptforartists.com/>