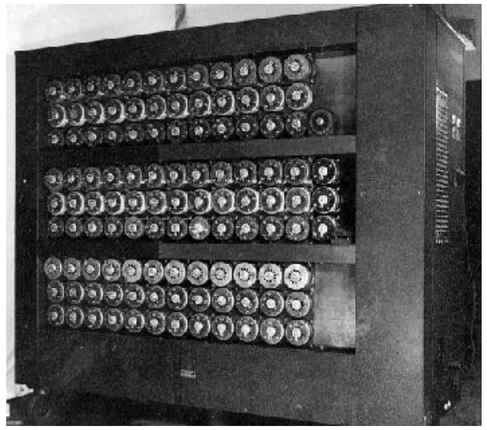
INTERACTION DESIGN ZHDK

MACHINE LEARNING

Physical Computing HS21

TIMELINE

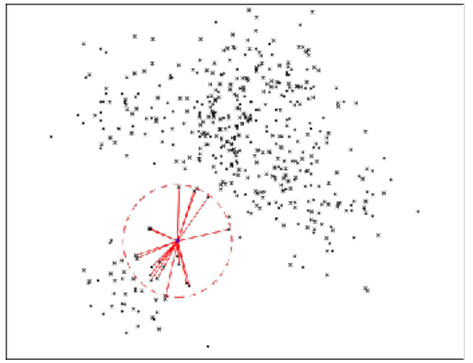


1950
Turing Test

1952
Checkers

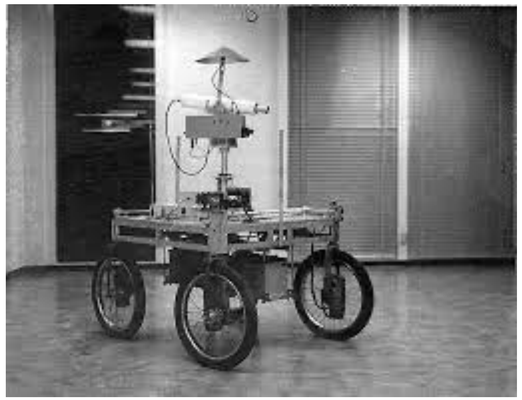
1957
The Perceptron

1967
The Nearest Neighbour



TIMELINE

1974-1980
AI WINTER



1979
The
Stanford
Cart

1996
IBM Deep Blue

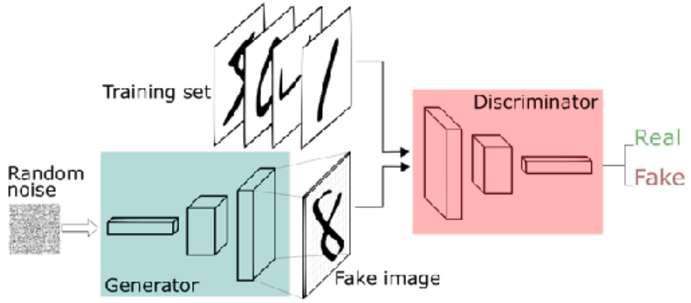
2002
Torch

TIMELINE

2006
Deep Learning

2011
IBM Watson

2012
ImageNet



2014
GANs

A Fast Learning Algorithm for Deep Belief Nets

Geoffrey E. Hinton
hinton@cs.toronto.edu
Simon Osindero
osindero@cs.toronto.edu
Department of Computer Science, University of Toronto, Toronto, Canada M5S 3G4

TIMELINE

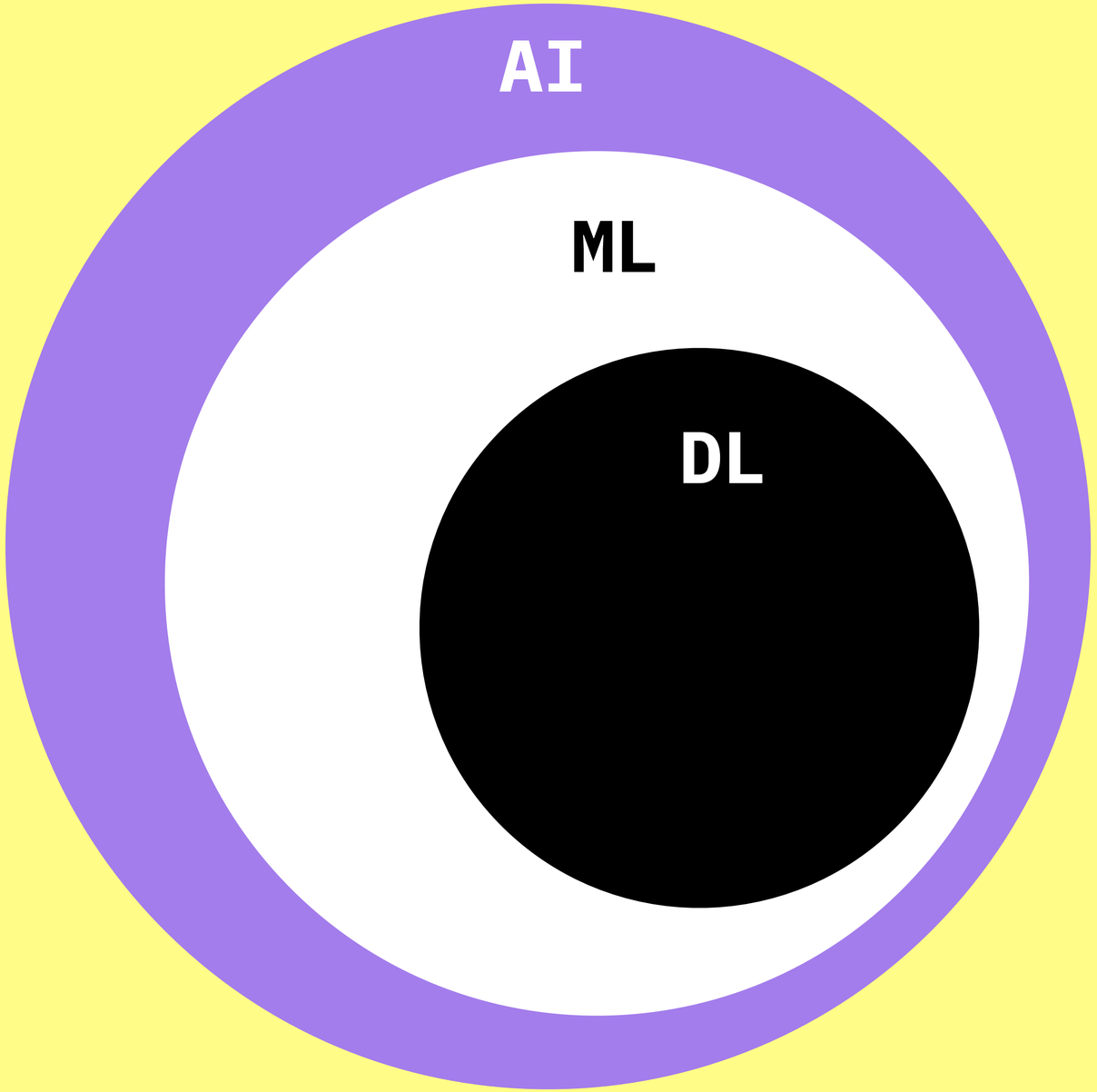
2016
AlphaGo

2017
Deepfake

2018
Tensorflow.js
ml5.js

ml5

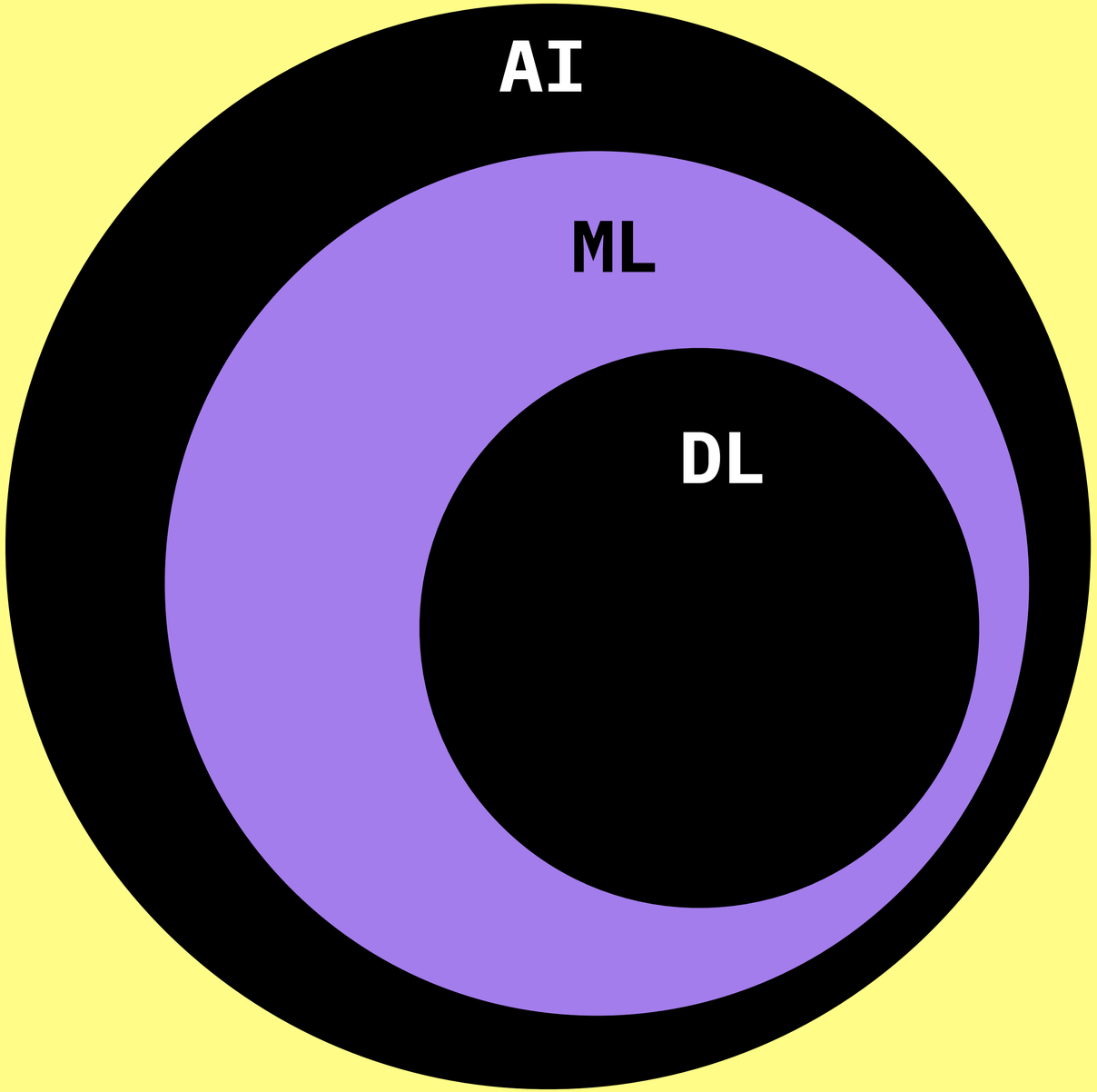
?



ARTIFICIAL INTELLIGENCE*

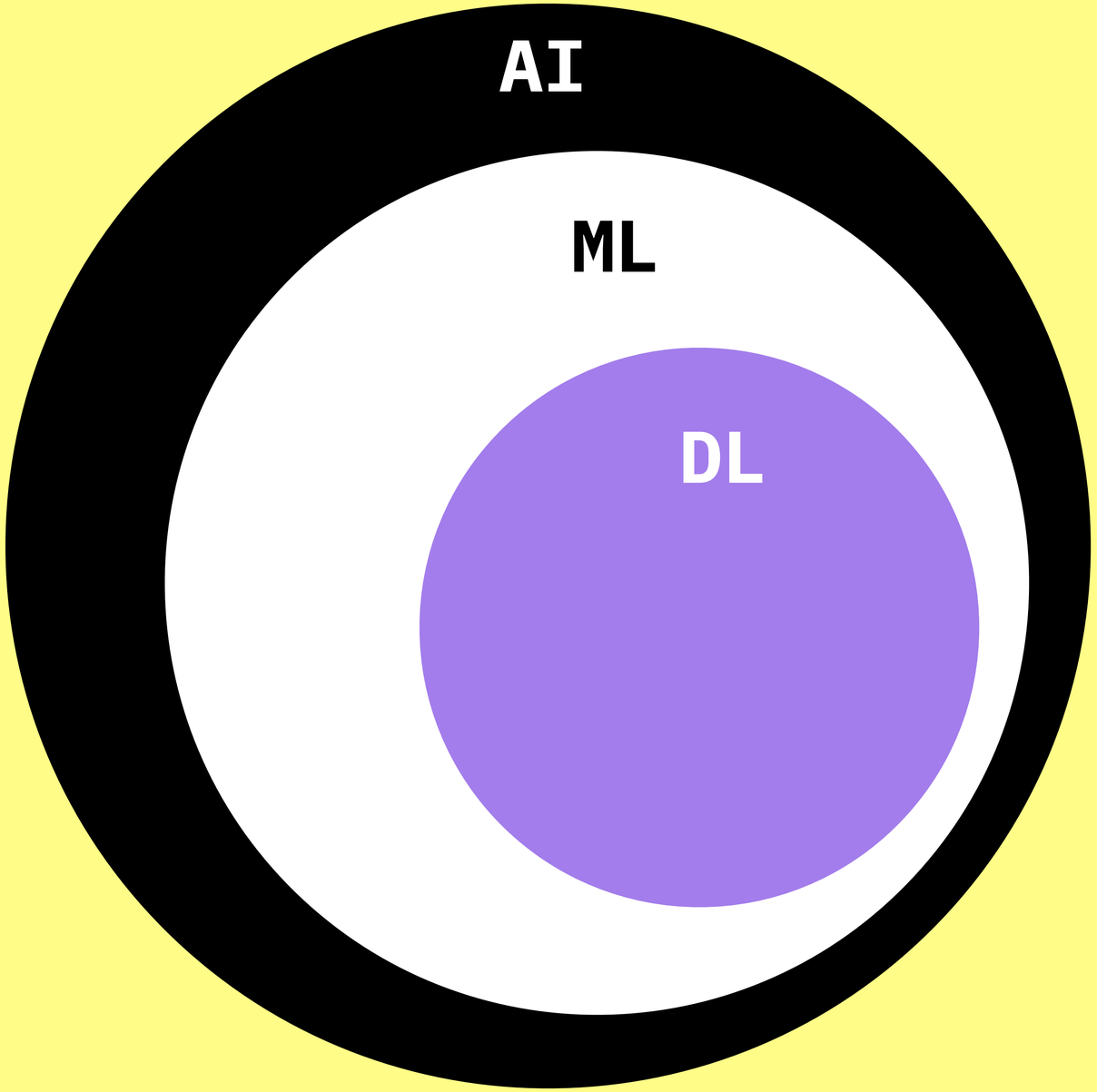
Technologies that are able to **perform specific tasks** as well as, **or better than**, we humans can. Today most AI is a form of Narrow AI - a system that can do one or a few defined things the same way or better than a human.

*Narrow AI



MACHINE LEARNING

Uses **statistical techniques** to give computer systems the ability to learn (i.e., progressively improve performance on a specific task) with data, **without being explicitly programmed.**



DEEP LEARNING

A technique for implementing ML, where the code structures are arranged in layers that loosely mimic the human brain.

USAGE

Pattern Recognition: Facial recognition, optical character recognition, etc.

Time Series Prediction: Will the stock rise or fall tomorrow? Will it rain or be sunny?

Signal Processing : Neural networks can be trained to process an audio signal and filter out unnecessary noise and amplify the important sounds

Control : Self-driving cars

Soft Sensors : Analyzing a collection of many measurements.Neural networks can be employed to process the input data from many individual sensors and evaluate them as a whole.

Anomaly Detection: Generate an output when something occurs that doesn’t fit the pattern.

https://natureofcode.com/book/chapter-10-neural-networks/

INTERACTION DESIGN ZHDK

TRAINING

Physical Computing HS21

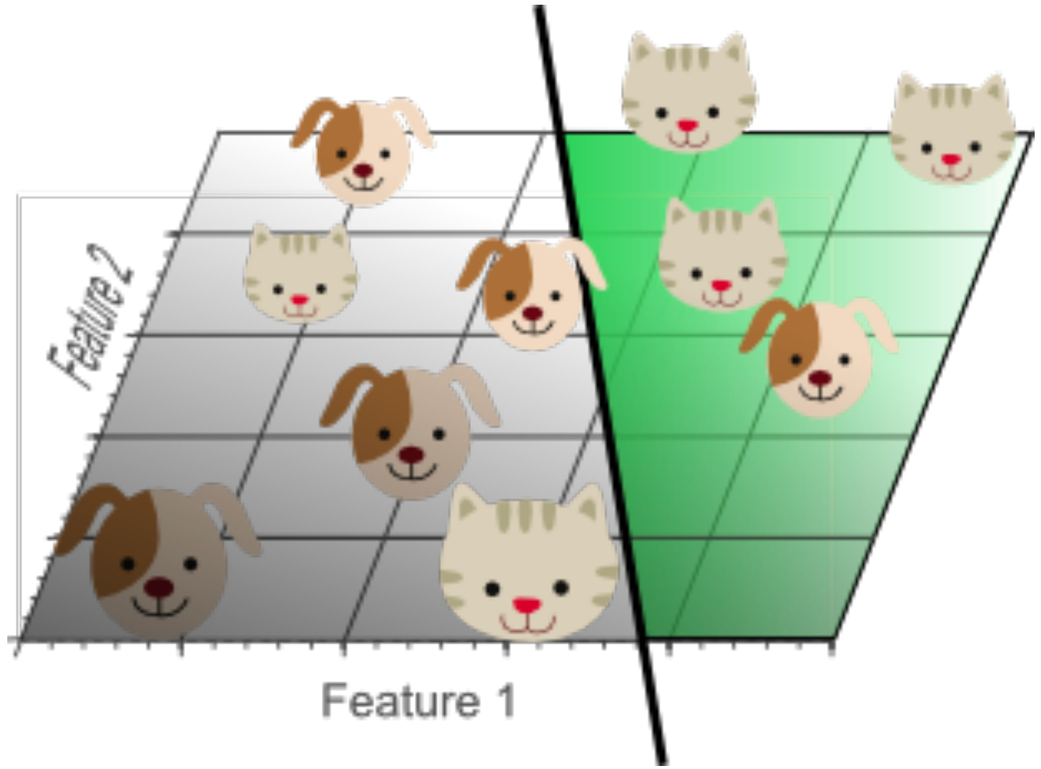
FEATURES/ATTRIBUTES

Features are used to train an ML system. They are the properties of the things you are trying to learn about, where one feature corresponds to one dimension.

The ML system can split the data and make future classifications with items that the system hasn’t seen before.

FEATURES/ATTRIBUTES

Features: Ear Length, Fur structure



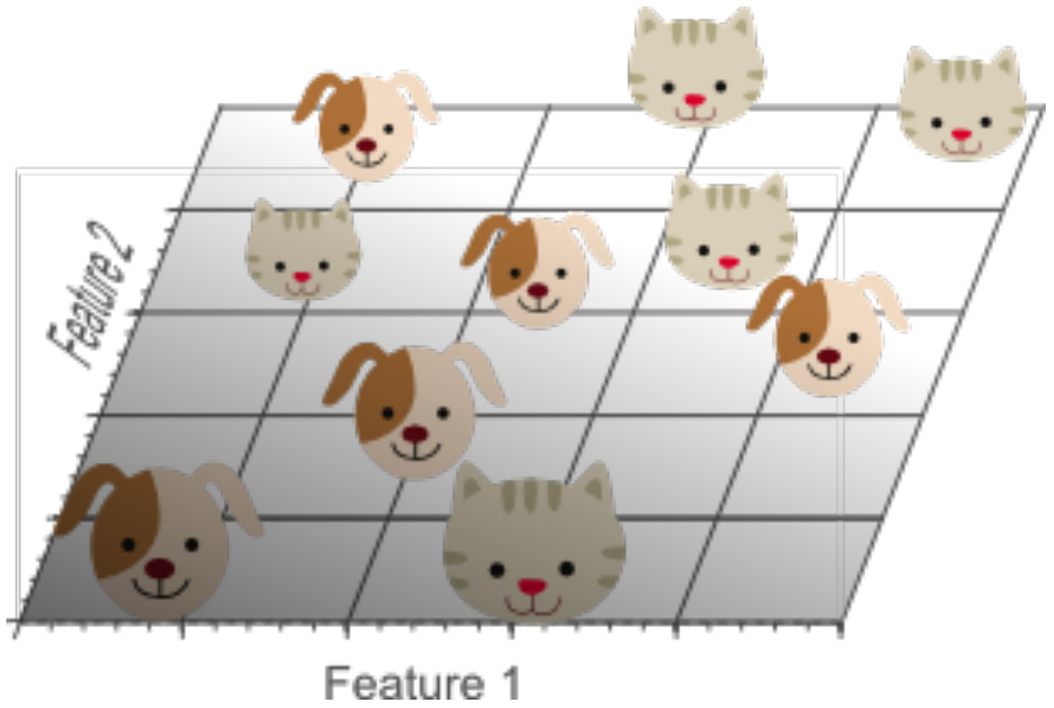
https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

FEATURES/ATTRIBUTES

Choosing the features has a big impact on the quality of the images. It thus important to choose the right features for each item.

FEATURES/ATTRIBUTES

Features: Fur Color, Weight



https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

LEARNING STYLES

Supervised learning

- regression: predict numerical values
- classification: predict category
- probability estimation: predict probability

The training data + the output we want to obtain (labels) are provided

Unsupervised learning

- clustering: group data according to "distance"
- association: find frequent co-occurrences
- link prediction: discover relationships in data
- data reduction: many features to fewer features

Data for training is provided, the desired output is not provided

Reinforcement learning

- Trial and error: achieve a goal in an uncertain, potentially complex environment

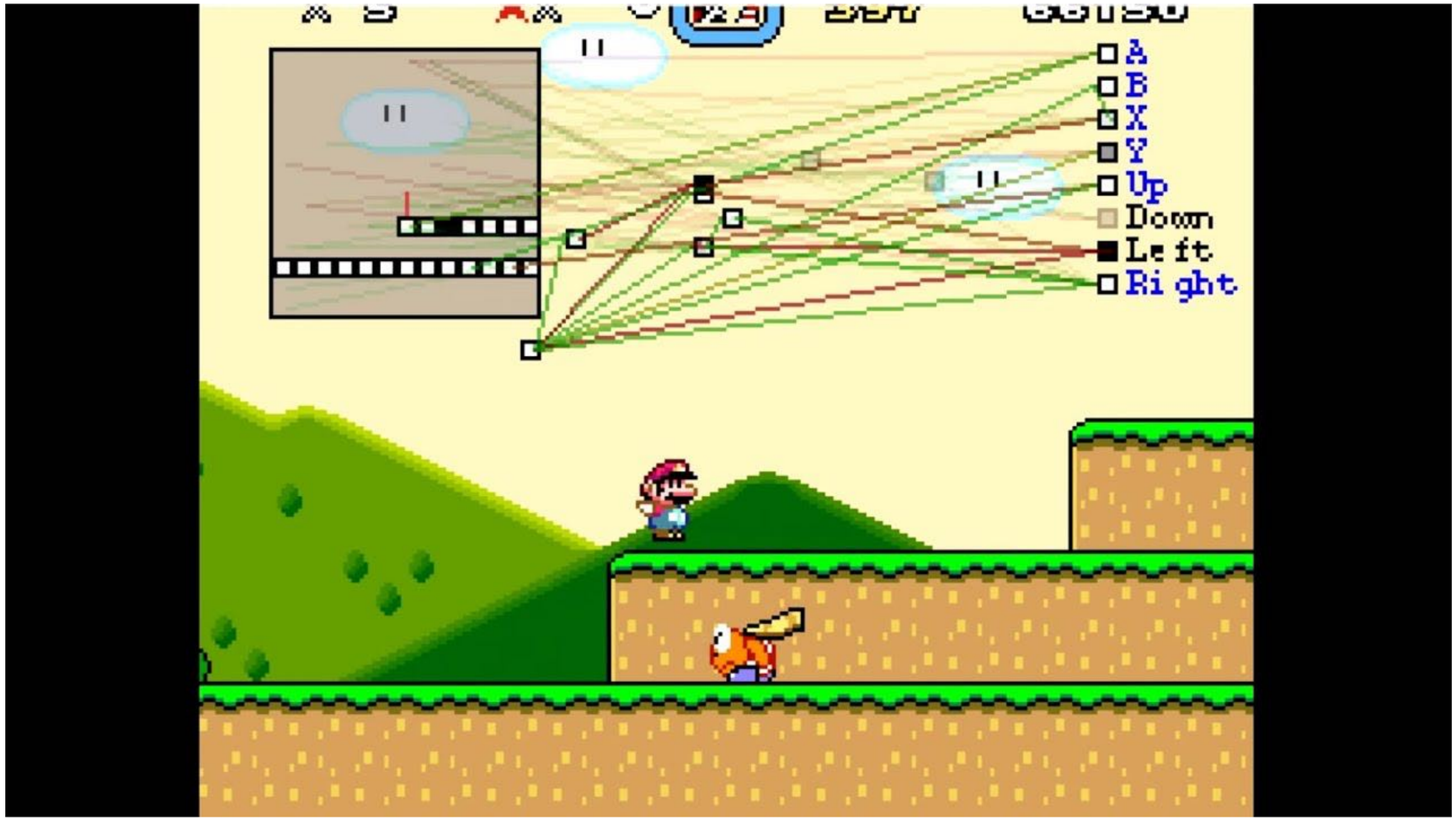
Rewards, when certain tasks are performed correctly are provided

SUPERVISED

https://editor.p5js.org/AndreasRef/sketches/H1L-KrzFQ

UNSUPERVISED

REINFORCMENT



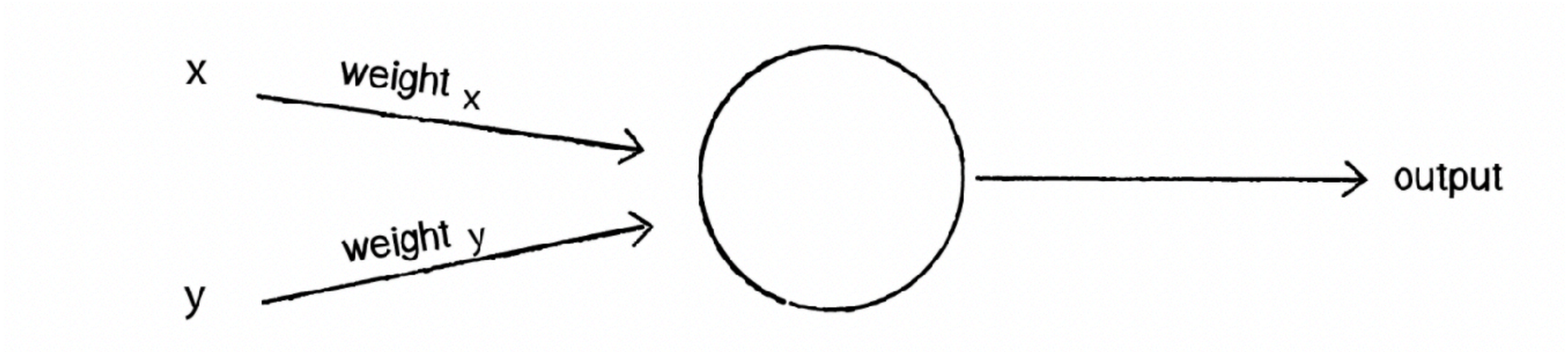
INTERACTION DESIGN ZHDK

NEURAL NETWORKS

Physical Computing HS21

(DEEP) NEURAL NETWORK

Invented in 1957 by Frank Rosenblatt at the Cornell Aeronautical Laboratory, a perceptron is the simplest neural network possible: a computational model of **a single neuron.**



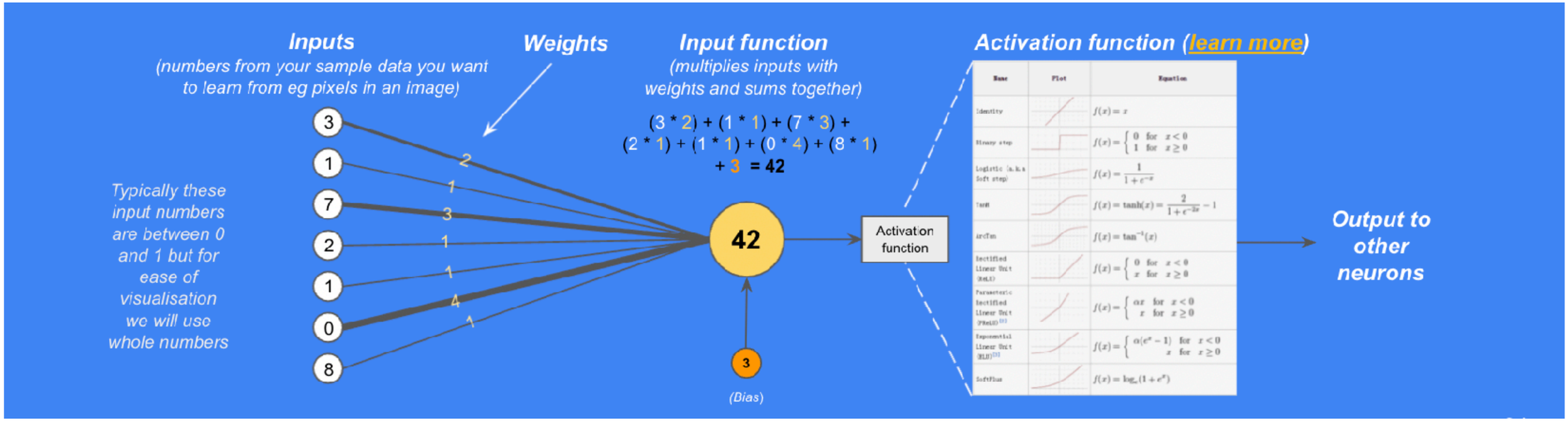
https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf

PERCEPTRON

A neuron has some **weighted inputs** that are summed together. A **bias** is then added to the total.

The weights and bias are determined when we train the system.

If final result is greater than a **threshold**, it activates the activation function and the output is forwarded into other neuron, where the process repeats.



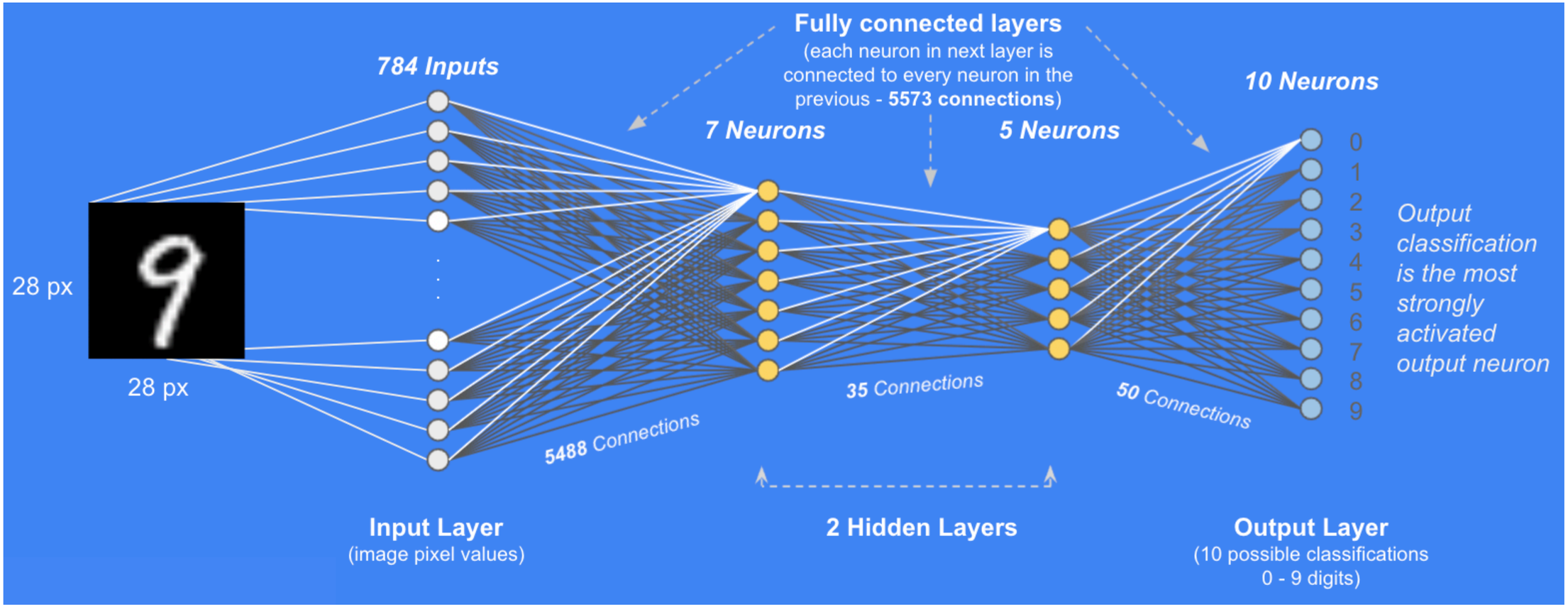
https://docs.google.com/presentation/d/1kSuQyW5DTnkVaZEjGYCkfOxvzCqGEFzWBy4e9Uedd9k/edit#slide=id.g283eef556d_0_307

PERCEPTRON

TENSORFLOW VISUALISATION

NEURAL NETWORKS

Neural networks are composed of layers of Perceptrons, with connections in different layers. These networks transform data until they can classify it as an output.



https://docs.google.com/presentation/d/1kSuQyW5DTnkVaZEjGYCkfOxvzCqGEFzWBy4e9Uedd9k/edit#slide=id.g283eef556d_0_307

SUMMA TECHNOLOGIAE

INVISIBLE CITIES

NEW DARK AGE

PROFILES OF THE FUTURE

EXCAVATING.AI

NEURAL NETWORKS HISTORY

CULTURE MACHINE VOL.20

A QUICK GUIDE TO NEURAL NETWOK ARCHITECTURES

PAPER ON HCI VS AI

HOW TO FOOL AN ML ALGORITHM?

JIGSAW PROJECT

AI DUNGEON

BODY MOVEMENT LANGUAGE

OBJECTIFIER

THE POLITICS OF IMAGES IN MACHINE LEARNING TRAINING SETS

COLLECTION ON .JS-BASED AI APPS

CLASSIFYING WES ANDERSON MOVIES BY COLORS

THE ALCHEMY AND SCIENCE OF MACHINE LEARNING FOR GAMES

RECENT ETH PROJECT WITH FAST FLYING DRONES

TENSORFLOW.JS BASICS

DON’T DRAW A PENIS

MECHANICAL TURK

SERIES ON VIDEOS ON DEEP LEARNING

HOW TO USE T-SNE

GREAT INTRODUCTION TO TENSORFLOW

TEACHABLE MACHINE

DRIES DEPOORTER

RUNWAYML

OPEN-SOURCE INTRODUCTION TO MACHINE LEARNING

DESCRIPT

SOUGWEN CHUNG

OPEN.AI

QUASIMODO

ML5.JS THE CODING TRAIN

DATASET COLLECTION

REFIK ANADOL

AUDIO DATASET TRAINING

ANDREAS REFSGAARD

MEMO AKTEN

OPEN MV

OPENCV AI KIT

ANNA RIDLER

- BOOKS
- PROJECTS
- TOOLS
- ARTISTS
- ARTICLES/PAPERS
- GUIDES

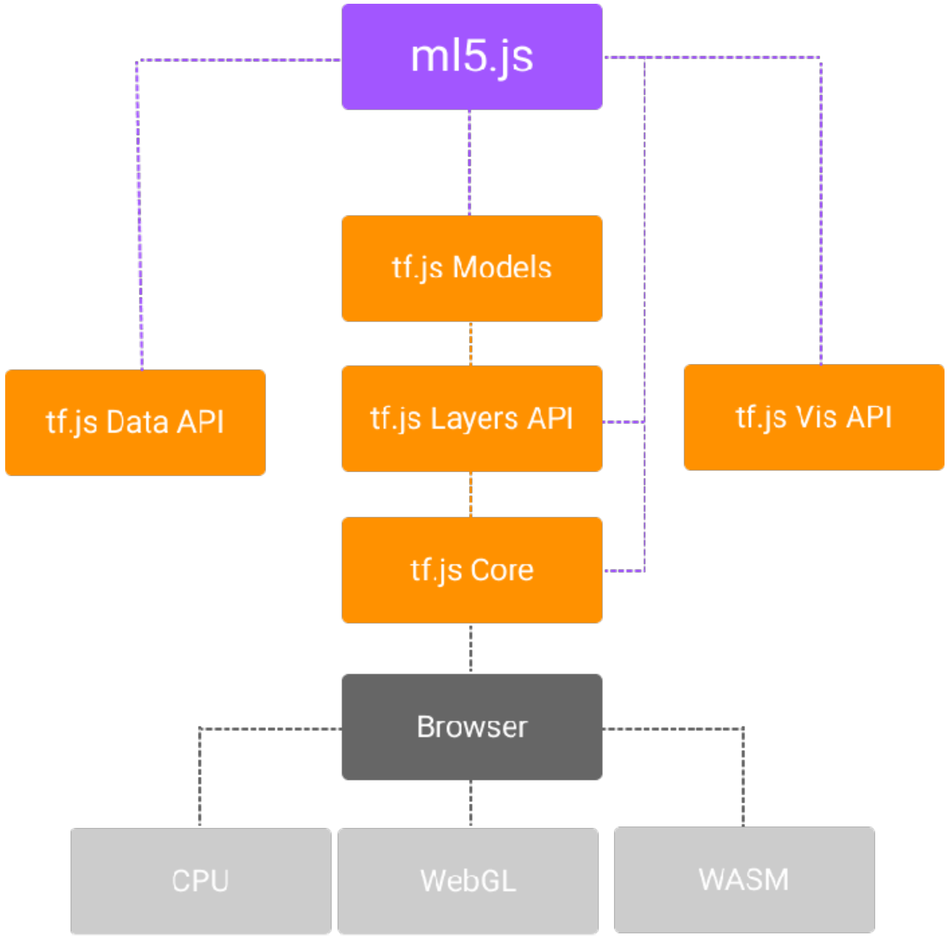
INTERACTION DESIGN ZHDK

ML5.JS

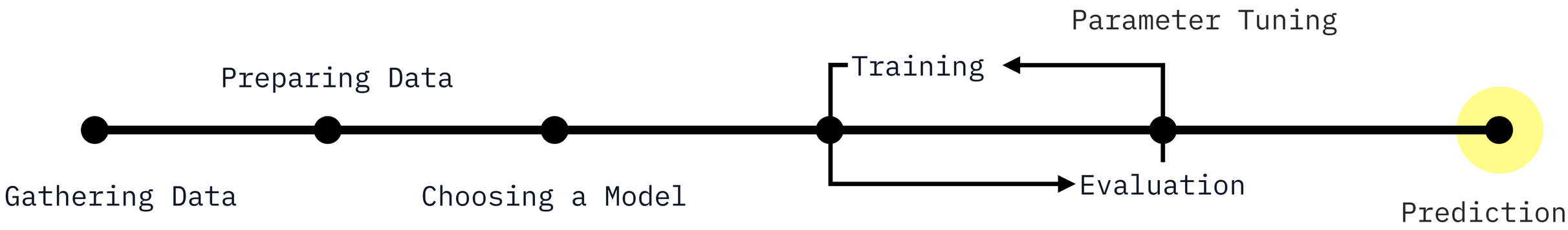
MUI HS21

ML5.JS

ml5.js provides access to machine learning algorithms and models in the browser, building on top of **TensorFlow.js**.



LEARNING STEPS



MODELS

IMAGE

imageClassifier
ObjectDetector
poseNet
BodyPix
UNET
YOLO
StyleTransfer
Pix2Pix
CartoonGAN
FaceApi
Facemesh
Handpose
CVAE
DCGAN
SketchRNN

SOUND

Pitch Detection
SoundClassifier

TEXT

CharRNN
Word Vectorization
Sentiment
UniversalSentenceEncoder

HELPERS

NeuralNetwork
FeatureExtractor
KNNClassifier
KMeans

MODELS

imageClassifier('MobileNet')

```
const classifier = ml5.imageClassifier('MobileNet');

classifier.classify(video, gotResult);

function gotResult(error, result) {
  console.log(result);
}
```

My guess is a toaster.

My confidence is 0.12.

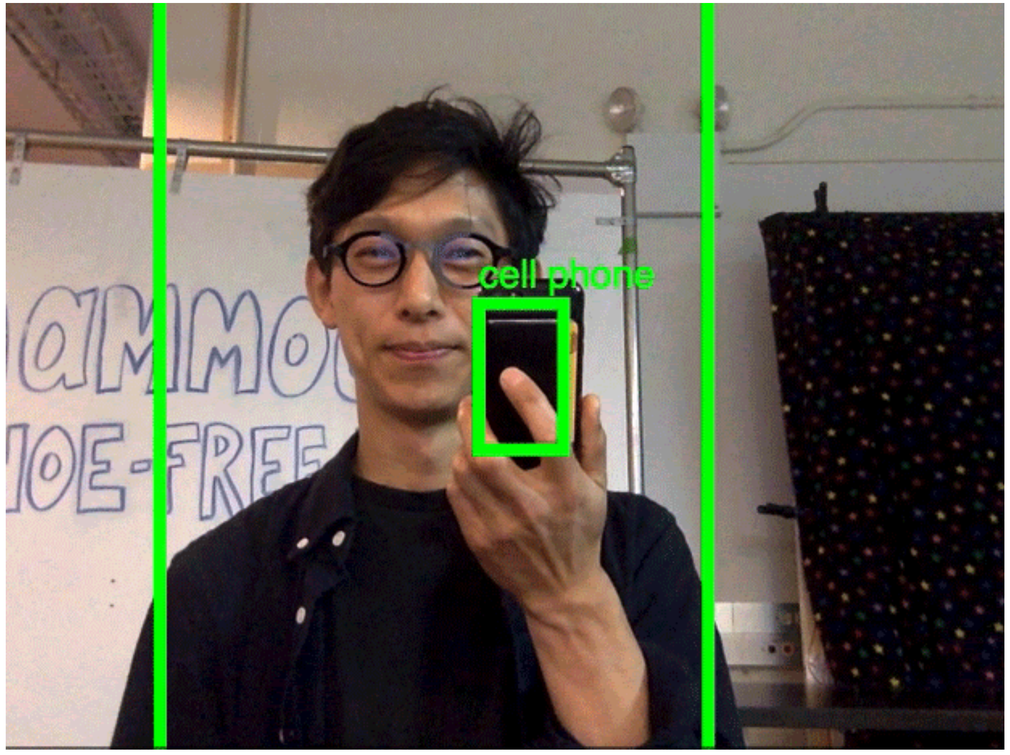
MODELS

ObjectDetector (YOLO or CocoSSd)

```
const classifier = ml5.imageClassifier('MobileNet');

classifier.classify(video, gotResult);

function gotResult(error, result) {
  console.log(result);
}
```



MODELS

PoseNet

```
const posenet = ml5.poseNet(video);

posenet.on('pose', function(results) {
  poses = results;
});

function draw() {
  if (poses.length > 0) {
    circle(poses[0].nose.x, poses[0].nose.y);
  }
}
```

MODELS

SoundClassifier('SpeechCommands18w')

Label: zero
Confidence: 0.99

```
let classifier = ml5.soundClassifier('SpeechCommands18w');

classifier.classify(gotResult);

function gotResult(error, result) {
  labelDiv.html(result[0].label);
  confidenceDiv.html(result[0].confidence);
}
```

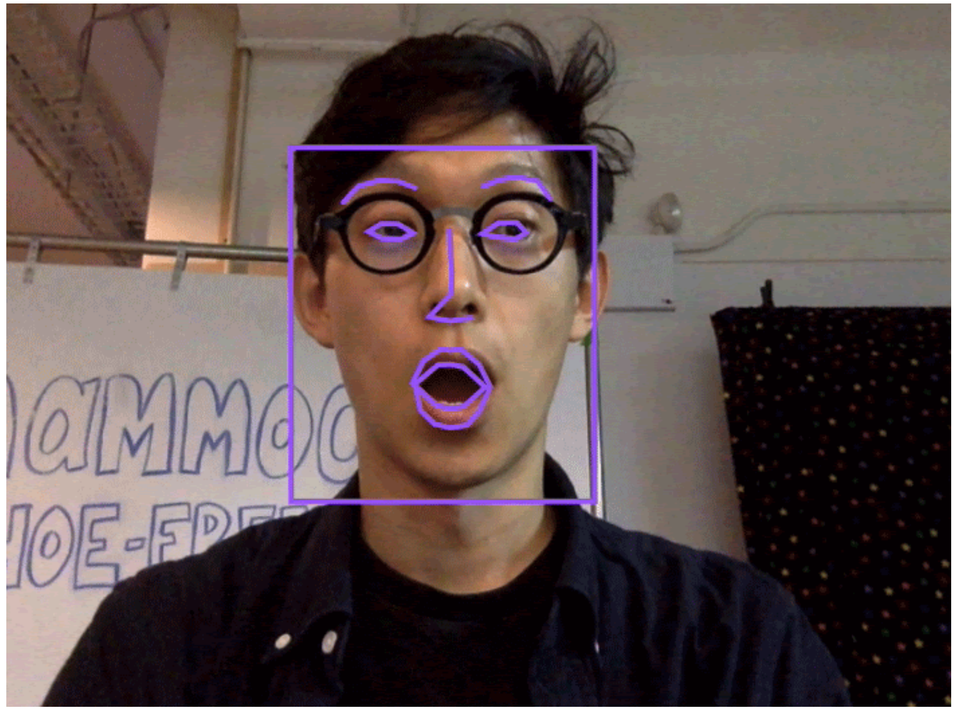
MODELS

Face-Api

```
const faceapi = ml5.faceApi();

faceapi.detect(video, gotResults);

function gotResults(error, results) {
  drawLandmarks(results);
}
```



MODELS

KNNClassifier + FeatureExtractor

```
const knnClassifier = ml5.KNNClassifier();

const featureExtractor =
ml5.featureExtractor('MobileNet', modelReady);

const features = featureExtractor.infer(myImg);

knnClassifier.addExample(features, label);

knnClassifier.classify(features, (err, result) => {
  console.log(result);
});
```

FeatureExtractor(mobileNet model) Loaded

Load Dataset

If you load this sample classifer dataset. Try to make rock, paper, or scissor gestures to see if the classifier can class them.
If this sample dataset doesn't work well for you, you could train your own classifier,
and use the 'Save Dataset' button below to create your own myKNNDataset.json file,
and replace the myKNNDataset.json in this folder.

✊ Add an Example to Class Rock | Reset Class Rock | 12 Rock examples | Confidence in Rock is: 100 %

🖐 Add an Example to Class Paper | Reset Class Paper | 13 Paper examples | Confidence in Paper is: 0 %

✌ Add an Example to Class Scissor | Reset Class Scissor | 16 Scissor examples | Confidence in Scissor is: 0 %

Start predicting!

Clear all classes

KNN Classifier with mobileNet model labeled this as Class: Rock with a confidence of 100 %

Save Dataset