Artificial intelligence

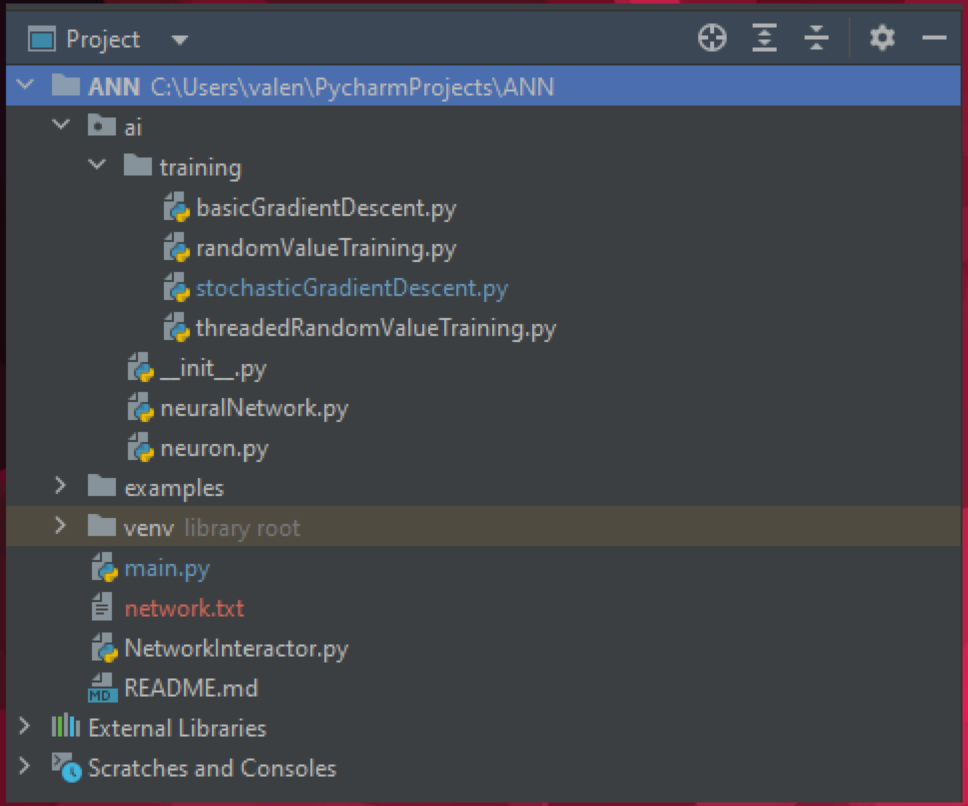
Heiderich Valentin

2023

Content

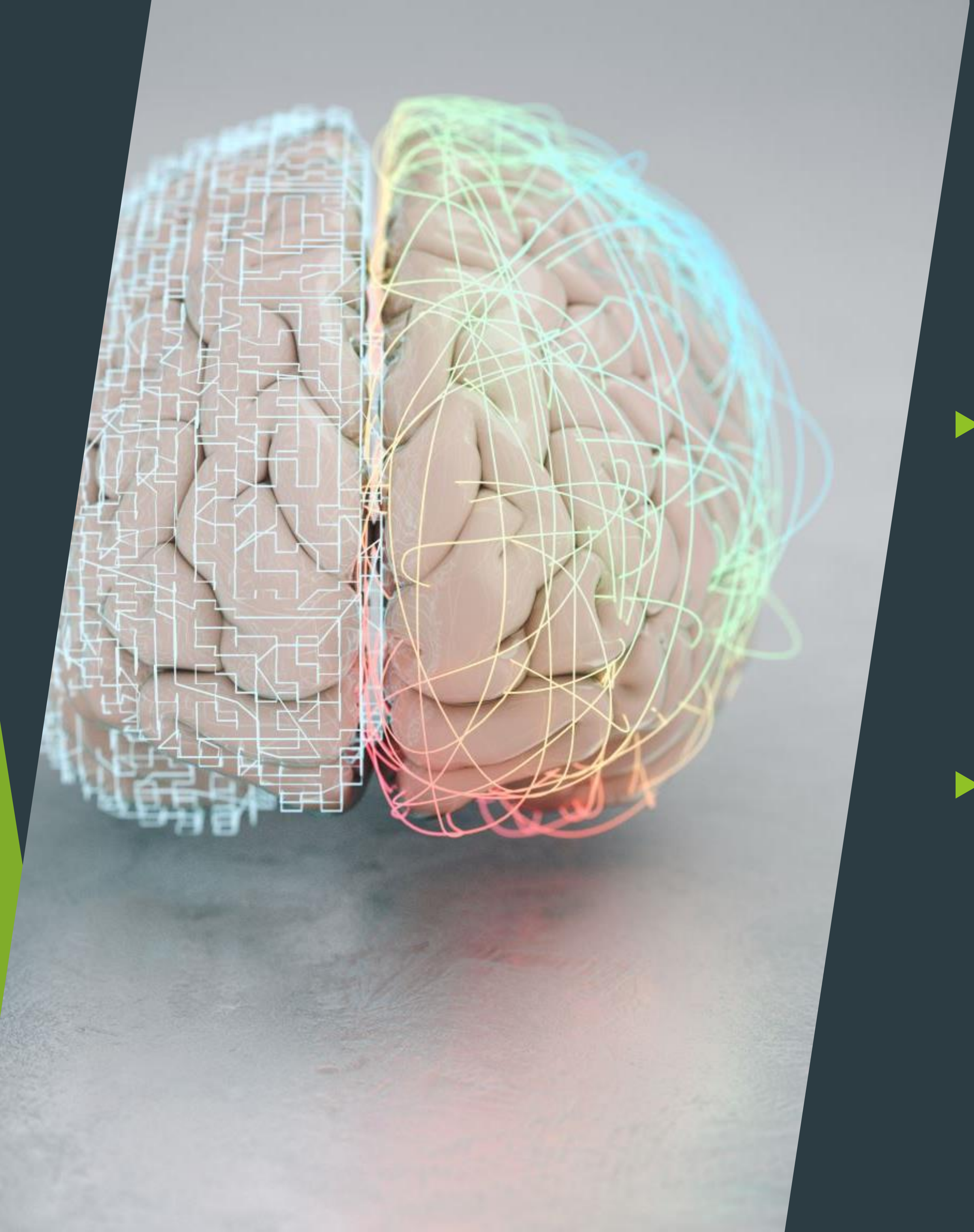
- The Theory
  - Code Example (Don’t worry, you don’t have to understand)
  - The Human Brain
  - Neural networks in programming
- Development of AI over time
  - AI applications
  - Neural Networks
- Use Cases
  - Cars
  - Chat bots
  - Biometric data collection
- Social Aspects, Politics and Law
  - AI Act
  - Data
- Conclusion

The Theory


Project
ANN C:\Users\valen\PycharmProjects\ANN
ai
training
basicGradientDescent.py
randomValueTraining.py
stochasticGradientDescent.py
threadedRandomValueTraining.py
__init__.py
neuralNetwork.py
neuron.py
examples
venv library root
main.py
network.txt
NetworkInteractor.py
README.md
External Libraries
Scratches and Consoles

```python
import random
import math


1 usage  Valentin *
def sigmoid(output):                                  # expanded activation function
    try: return 1 / (1 + math.exp(-output))
    except OverflowError: return 0


10 usages  vh64g *
class neuron:
    vh64g *
    def __init__(self):
        self.weights = None  #all weights connected to the specific neuron
        self.costGradientWeights = [] # for gd training algorithms
        self.costGradientBias = None # for gd training algorithms
        self.bias = 1 # bias added to summed value
        self.output = None # init output

    5 usages (5 dynamic)   new *
    def randomize(self, input_count): # create random weights and a random bias, input count must equal len(cn)
        new_weights = [] # init nw var
        for i in range(input_count): new_weights.append(random.uniform(-10, 10)) # create a random weight (type: float) for each connection (between -10 and 10)
        self.bias = random.uniform(-10, 10) # create random bias (between -10 and 10)
        self.weights = new_weights # push weights

    11 usages (11 dynamic)   new *
    def calc(self, inputs): # calculating the output weight based on some input list len(inputs) must equal len(weights)!
        self.output = 0 # clearing output var
        for i in range(len(inputs)): self.output += inputs[i] * self.weights[i] # sum products of each input value multiplied with the corresponding weight
        self.output += self.bias # adding bias value
        self.output = sigmoid(self.output) # using a sigmoid func on the output for clamping
        return self.output # return output
```

```python
class artificialNeuralNetwork:
    vh64g *
    def __init__(self, input_layer, output_layer, hidden_layers=None):

        self.input_layer = input_layer  # structure of input layer: [neuron1, neuron2, ...]
        self.hidden_layers = hidden_layers  # structure of hidden layers: [[layer1_neurons], [layer2: neurons, ...]
        self.output_layer = output_layer  # structure of output layer: [neuron1, neuron2, ...]

        if hidden_layers is None: self.hidden_layers = [] # avoid null pointer exceptions

        self.out = [] # init output value

        self.randomize()

    6 usages (5 dynamic)  vh64g +1 *
    def randomize(self):
        for layer in self.hidden_layers: # call the randomize function of each neuron in each hidden layer and the output layer, passing the len(inputs)
            if self.hidden_layers.index(layer) == 0:
                for neuron in self.hidden_layers[self.hidden_layers.index(layer)]:
                    neuron.randomize(len(self.input_layer))
            else:
                for neuron in self.hidden_layers[self.hidden_layers.index(layer)]:
                    neuron.randomize(len(self.hidden_layers[self.hidden_layers.index(layer) - 1]))
        for neuron in self.output_layer:
            neuron.randomize(len(self.hidden_layers[-1]))

    11 usages (11 dynamic)  vh64g *
    def calc(self, inputs): # calculating the output of the entire neural network
        self.input_layer = inputs # getting inputs
        self.out = [] # init output: type: list
        for layer in self.hidden_layers: # loop through each hidden layer
            for neuron in self.hidden_layers[self.hidden_layers.index(layer)]: # loop through each neuron in hidden layer
                if self.hidden_layers.index(layer) == 0: x = neuron.calc(self.input_layer) # calculating each neuron output of h0 based on input values
                else: x = neuron.calc([neuron.output for neuron in self.hidden_layers[self.hidden_layers.index(layer) - 1]])
                # calculating the other neurons, using the output values of the last layer
        for neuron in self.output_layer:
            middle_outs = [neuron.output for neuron in self.hidden_layers[-1]] # calculating the output for the output layer
            self.out.append(abs(neuron.calc(middle_outs))) # push output
        return self.out # return output
```



The Brain

- Neurons:

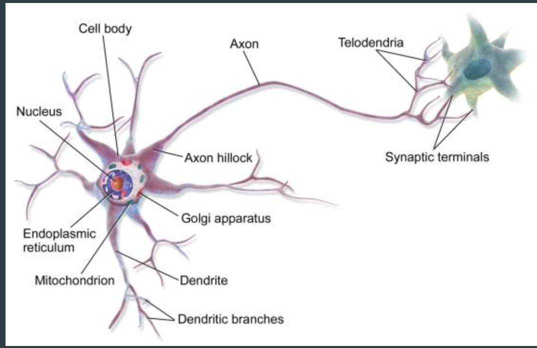
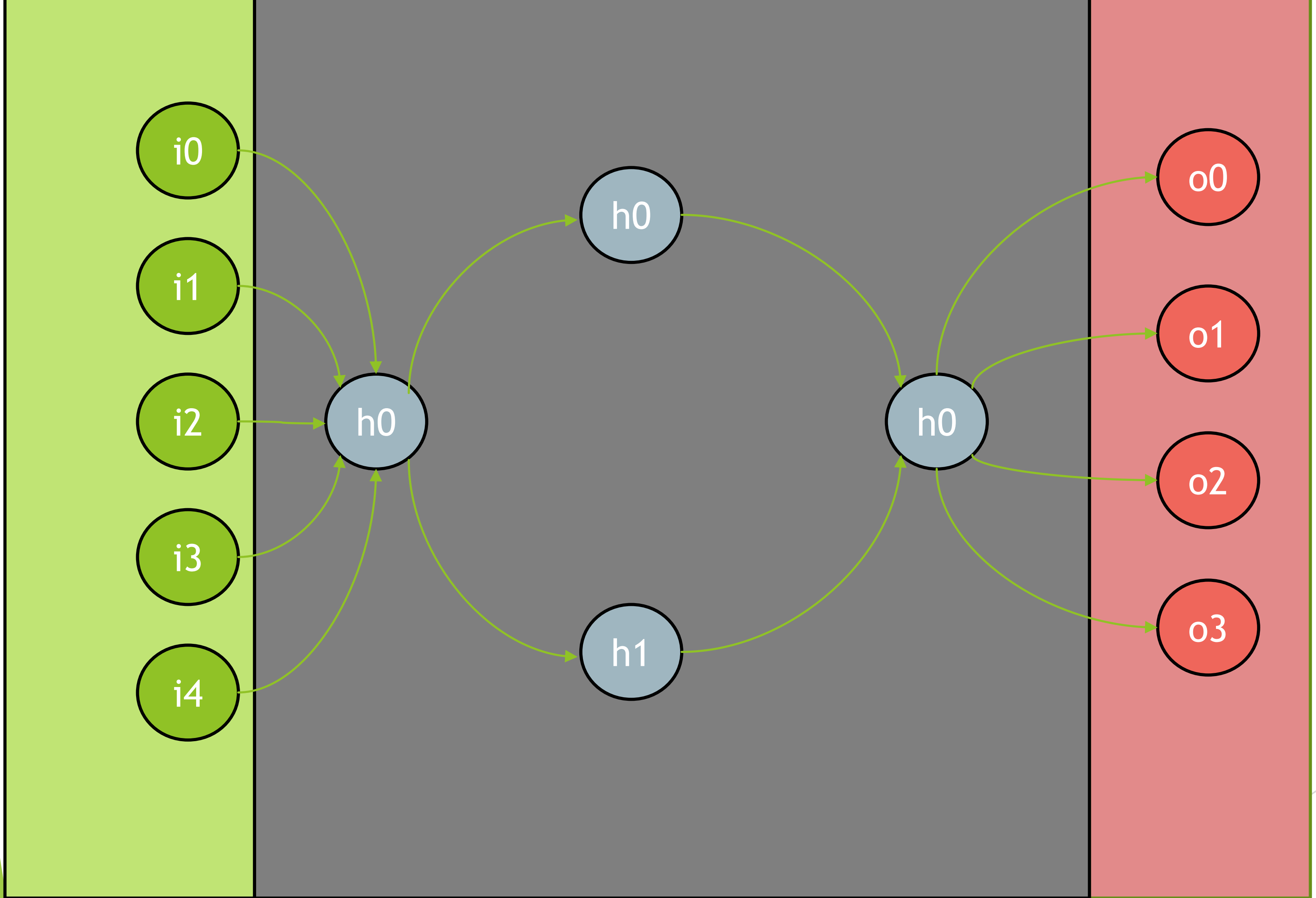


Image URL

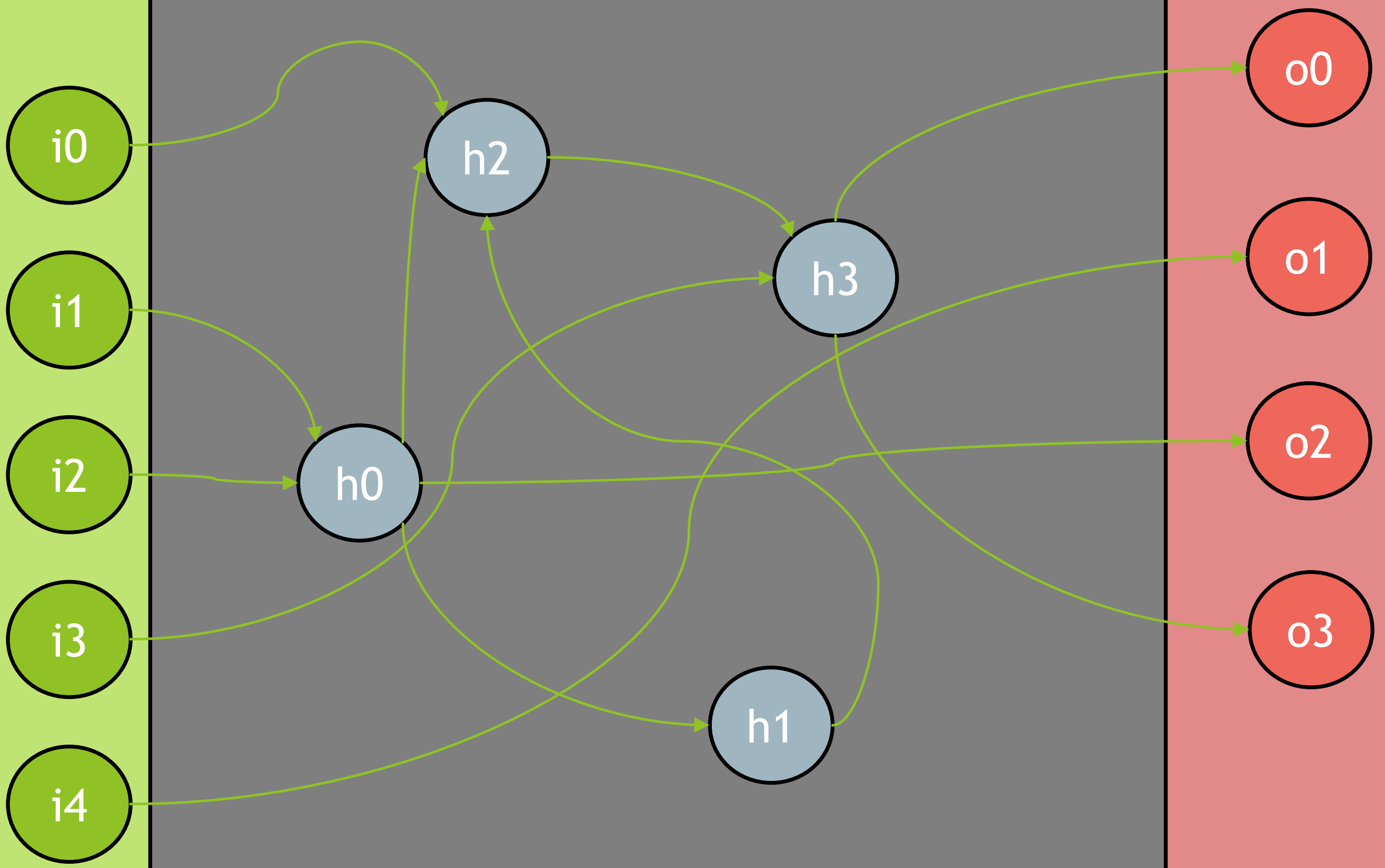
- Neural Networks:

Image URL

Neural Networks in programming

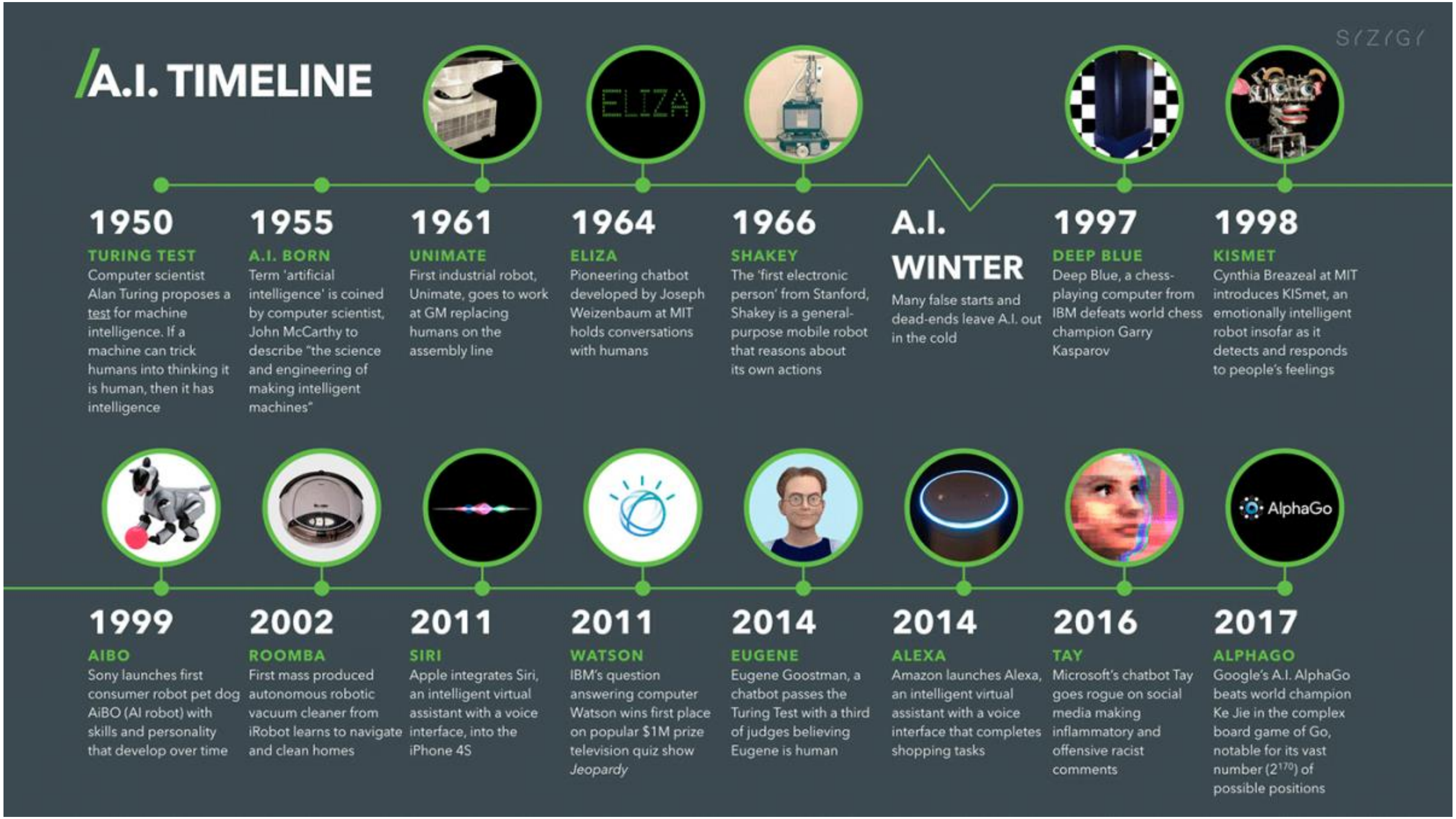


Neural Networks in programming



https://www.youtube.com/watch?v=N3tRFayqVtk

The development of AI over time



SYZYGY
/A.I. TIMELINE
ELIZA
1950
TURING TEST
Computer scientist Alan Turing proposes a test for machine intelligence. If a machine can trick humans into thinking it is human, then it has intelligence
1955
A.I. BORN
Term 'artificial intelligence' is coined by computer scientist, John McCarthy to describe "the science and engineering of making intelligent machines"
1961
UNIMATE
First industrial robot, Unimate, goes to work at GM replacing humans on the assembly line
1964
ELIZA
Pioneering chatbot developed by Joseph Weizenbaum at MIT holds conversations with humans
1966
SHAKEY
The 'first electronic person' from Stanford, Shakey is a general-purpose mobile robot that reasons about its own actions
A.I. WINTER
Many false starts and dead-ends leave A.I. out in the cold
1997
DEEP BLUE
Deep Blue, a chess-playing computer from IBM defeats world chess champion Garry Kasparov
1998
KISMET
Cynthia Breazeal at MIT introduces KISmet, an emotionally intelligent robot insofar as it detects and responds to people's feelings
AlphaGo
1999
AIBO
Sony launches first consumer robot pet dog AiBO (AI robot) with skills and personality that develop over time
2002
ROOMBA
First mass produced autonomous robotic vacuum cleaner from iRobot learns to navigate and clean homes
2011
SIRI
Apple integrates Siri, an intelligent virtual assistant with a voice interface, into the iPhone 4S
2011
WATSON
IBM's question answering computer Watson wins first place on popular $1M prize television quiz show *Jeopardy*
2014
EUGENE
Eugene Goostman, a chatbot passes the Turing Test with a third of judges believing Eugene is human
2014
ALEXA
Amazon launches Alexa, an intelligent virtual assistant with a voice interface that completes shopping tasks
2016
TAY
Microsoft's chatbot Tay goes rogue on social media making inflammatory and offensive racist comments
2017
ALPHAGO
Google's A.I. AlphaGo beats world champion Ke Jie in the complex board game of Go, notable for its vast number (2^{170}) of possible positions

History of neural networks

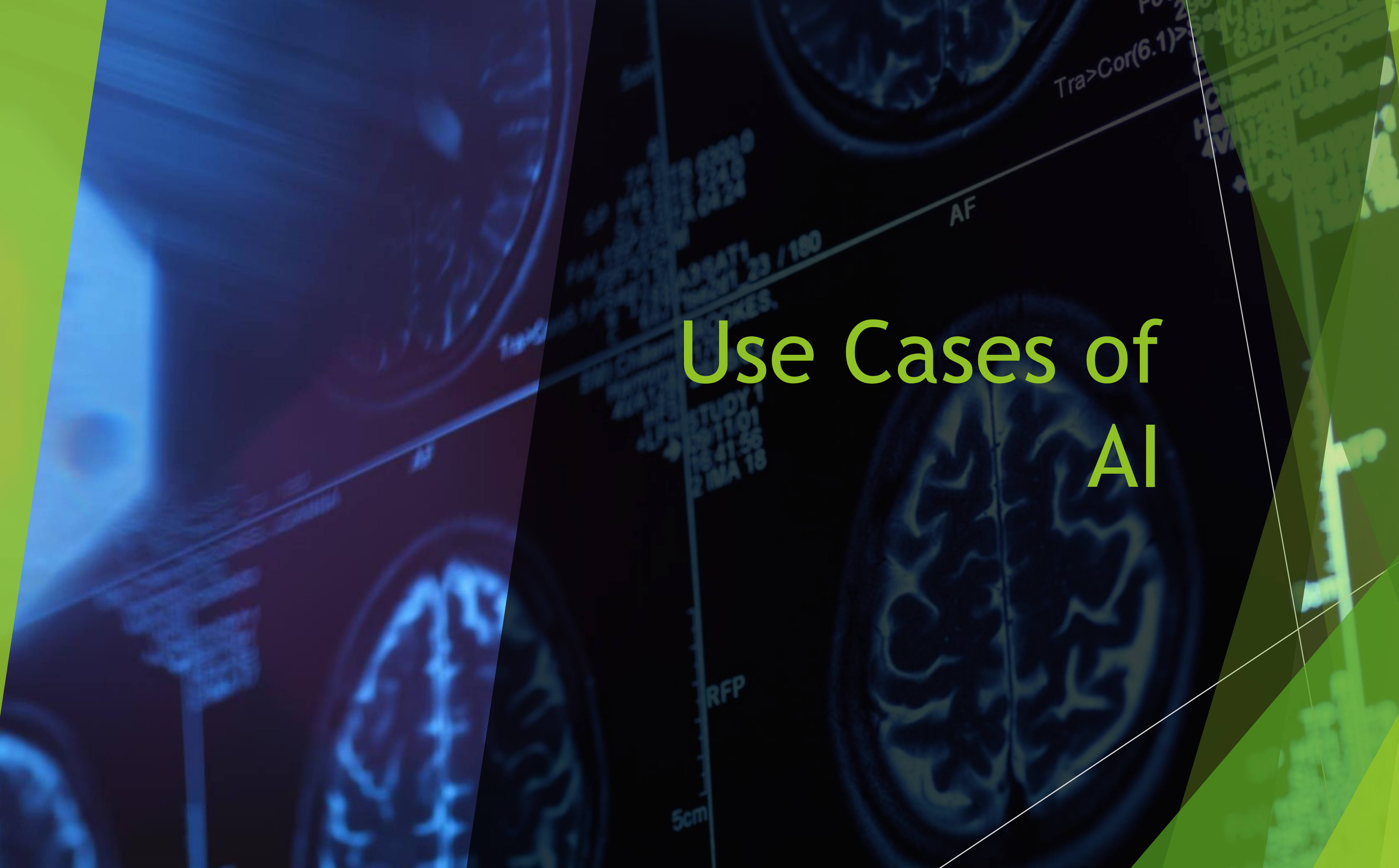
The history of neural networks is longer than most people think. While the idea of “a machine that thinks” can be traced to the Ancient Greeks, we’ll focus on the key events that led to the evolution of thinking around neural networks, which has ebbed and flowed in popularity over the years:

1943: Warren S. McCulloch and Walter Pitts published “A logical calculus of the ideas immanent in nervous activity (PDF, 1 MB) (link resides outside ibm.com)” This research sought to understand how the human brain could produce complex patterns through connected brain cells, or neurons. One of the main ideas that came out of this work was the comparison of neurons with a binary threshold to Boolean logic (i.e., 0/1 or true/false statements).

1958: Frank Rosenblatt is credited with the development of the perceptron, documented in his research, “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain” (link resides outside ibm.com). He takes McCulloch and Pitt’s work a step further by introducing weights to the equation. Leveraging an IBM 704, Rosenblatt was able to get a computer to learn how to distinguish cards marked on the left vs. cards marked on the right.

1974: While numerous researchers contributed to the idea of backpropagation, Paul Werbos was the first person in the US to note its application within neural networks within his PhD thesis (PDF, 8.1 MB) (link resides outside ibm.com).

1989: Yann LeCun published a paper (PDF, 5.7 MB) (link resides outside ibm.com) illustrating how the use of constraints in backpropagation and its integration into the neural network architecture can be used to train algorithms. This research successfully leveraged a neural network to recognize hand-written zip code digits provided by the U.S. Postal Service.



Use Cases of AI

Autonomous Driving

- Waymo (Google)
- When will it come?
  - Fast evolving technology
  - Conflicts with the law
  - Public interest
  - Could generate huge incomes for companys

Pros	Cons
Safety	Expensive
Comfortable	Who is responsible if sth. goes wrong
Efficiency	

Chatbots

- ELIZA
- Large Language Models:
  - GPT (Generative Pre-trained Transformer) (Open AI)
  - LaMDA (Language Model for Dialog Applications) (Google)
- Chat GPT, Google Bard, Bing AI Chatbot, Tai.ai …

PROS	CONS
Supporting humans, answering (private) questions	They can make mistakes, and spread false Information
Natural Conversations, a friend for lonely people.	Separated from the real life
Learning more about how language works.	

Biometric data collection

- Security Cameras with face detection
- Databases storing information about the citizens

Pros	Cons
Law enforcement (helps the police)	Privacy violations
Preventions of terrorism	

Social Aspects, Politics and Law

The data used to train AIs

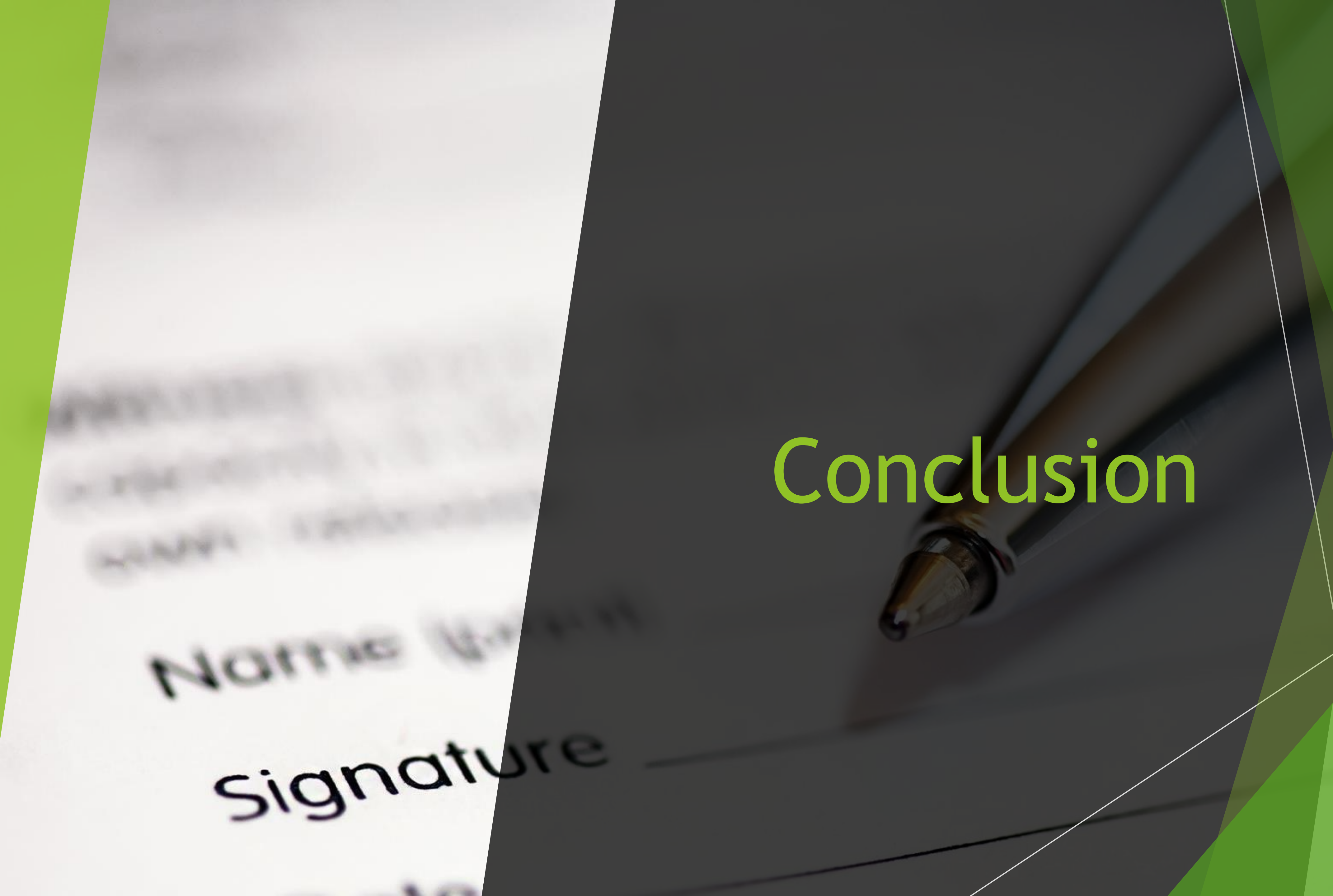
- Data gender gap!
  - Less data about women -> e.g., less or even wrong medical advice
- Racist data -> Racist AI

Example: Tai.ai

AI Act

- Proposed law for the European Parliament
- Aspects of the proposed law:
  - Liberal aspect of no face recognition
    - *„Gesichtserkennung zur Überwachung kennen wir aus China, diese Anwendung von Technologie hat in einer liberalen Demokratie nichts zu suchen. Es ist ein liberaler Gewinn und ein starkes Signal für die Verhandlungen mit den Mitgliedsländern, dass das Parlament sich für Bürgerrechte einsetzt und ein Verbot biometrischer Überwachung im öffentlichen Raum fordert.“ ~Svenja Hahn*
  - *European Union as Hotspot for the Development of AI*

Conclusion



AI has a big potential to help us

But we must be sure about AI used in Hospitals etc. that they are trained with a dataset representing all citizens

That’s a reason why the separation of different use cases for AI, regarding their purpose would be useful

AI also is a huge chance for new Start Ups, so we need to make the law as simple as possible while still protecting human rights .

Sources

- Event: Svenja Hahn visiting Stuttgart
- https://www.ibm.com/topics/neural-networks
- https://www.svenja-hahn.eu/post/pm-das-ki-gesetz-tragt-klare-liberale-handschrift-svenja-hahn-fdp-zur-heutigen-abstimmung-zur-parlamentsposition-beim-ai-act
- https://www.fdp.de/chancen-der-kuenstlichen-intelligenz-nutzen
- https://artificialintelligenceact.eu/
- https://artificialintelligenceact.eu/the-act/
- https://www.compliance-fit.com/post/die-vor-und-nachteile-von-chatbots-fur-websites

Image/Video Sources

- https://digitalwellbeing.org/wp-content/uploads/2017/08/Artificial-Intelligence-AI-Timeline-Infographic.jpeg
- https://www.youtube.com/watch?v=jTSn7f4sEKo&t=1s
- https://upload.wikimedia.org/wikipedia/commons/thumb/1/10/Blausen_0657_MultipolarNeuron.png/1200px-Blausen_0657_MultipolarNeuron.png
- https://news.harvard.edu/wp-content/uploads/2022/05/iStock-2500-1600x900.jpg