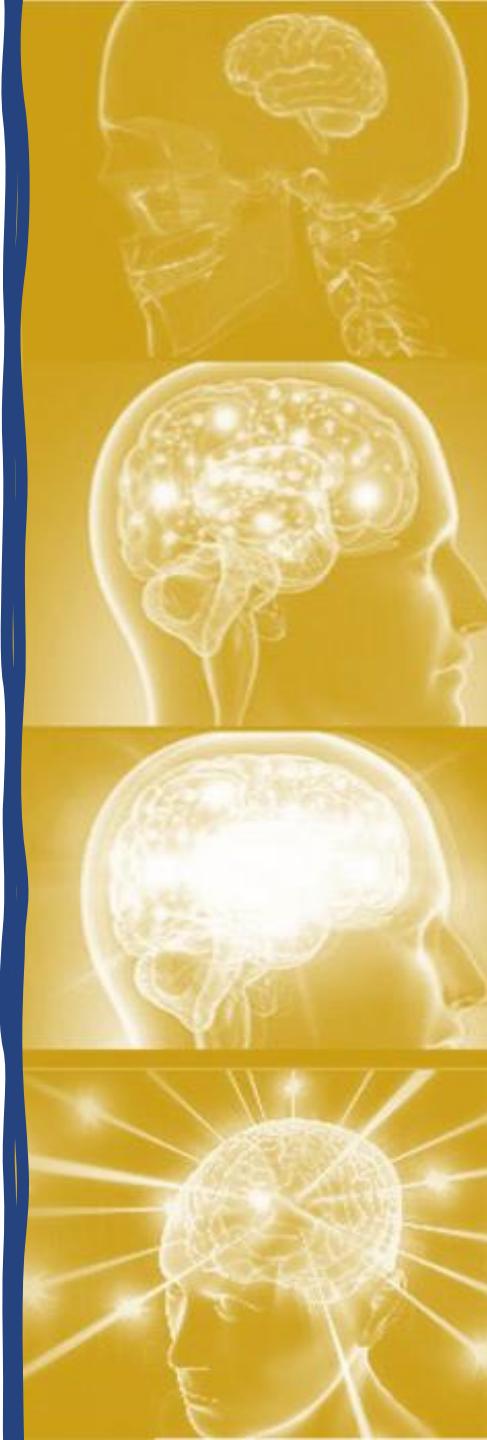


How Smart is AI Really?

Real projects to build
real understanding.

Renee Noble

 @noblerenee@mastodon.social
 reneenoble.com



Getting in trouble for
plagiarism when using
ChatGPT

Using AI to assist you
when doing repetitive
work.

Knowing Gen AI is not
"intelligent" & using it
in the defined contexts

Teaching others AI isn't
magic, it's maths! And
clarifying its strengths!

Hi, I'm Renee!



Tech +
Education +
Community!

Context driven learning experiences!

Have you heard about.....

AI

Have you heard about.....

AI

of course you have.

So, what is AI?

So, what is AI?

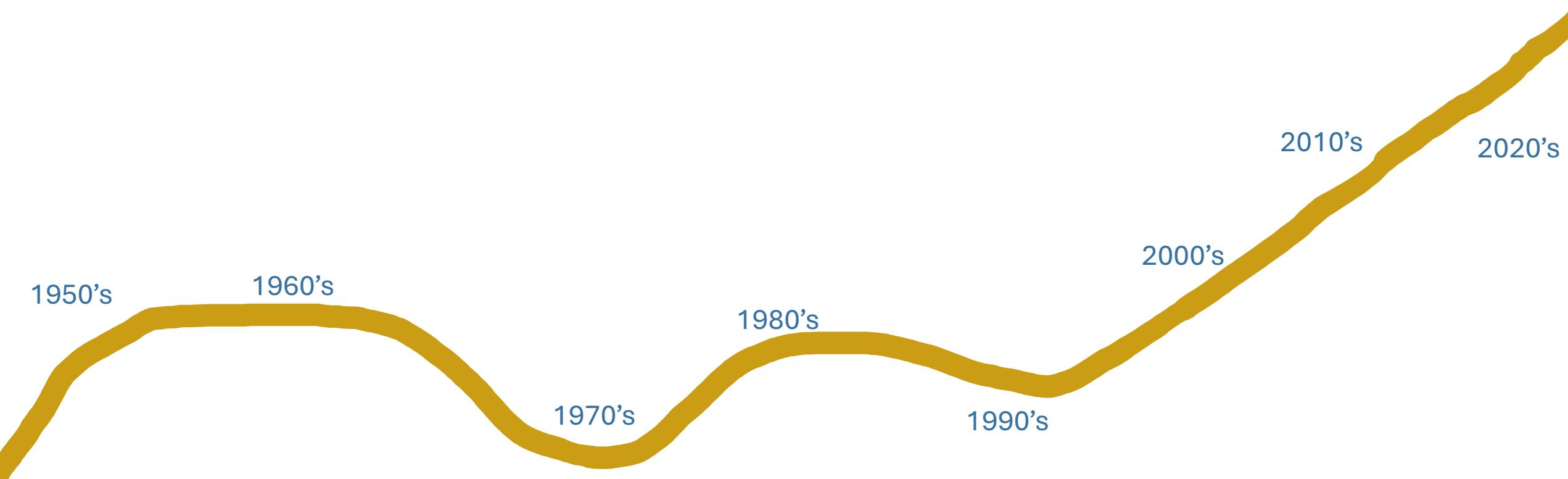
It's Artificial Intelligence

So, what is AI?

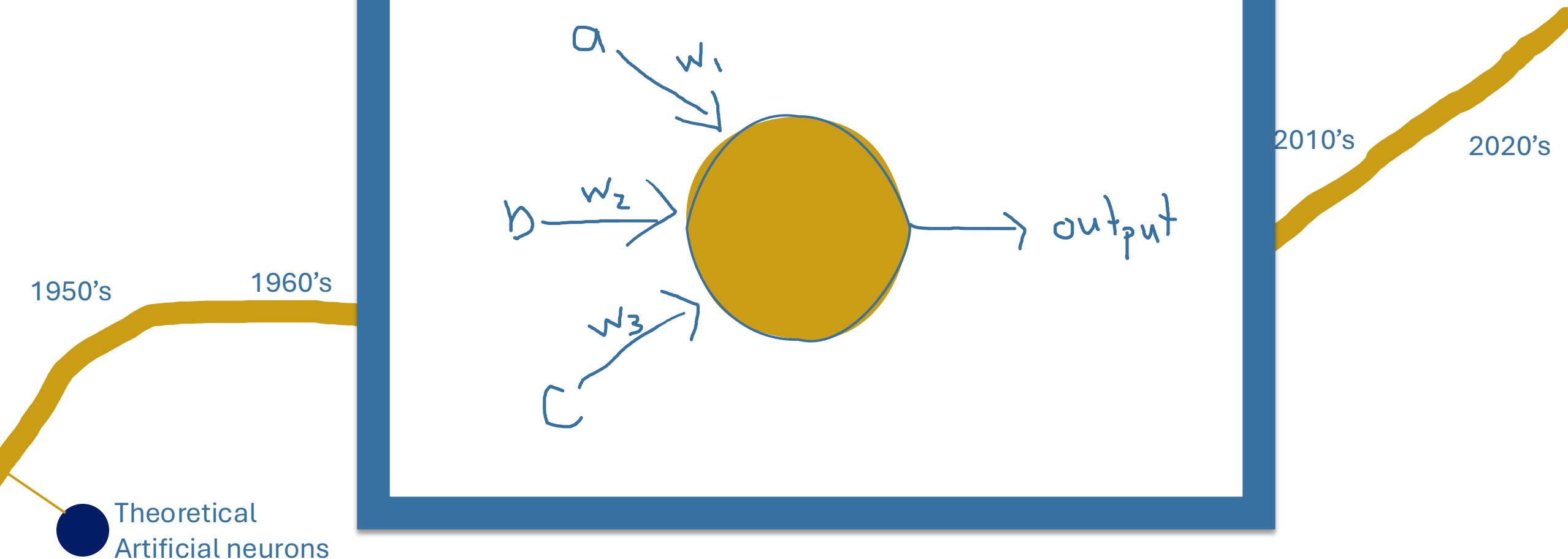
It's Artificial Intelligence

Done... Right?

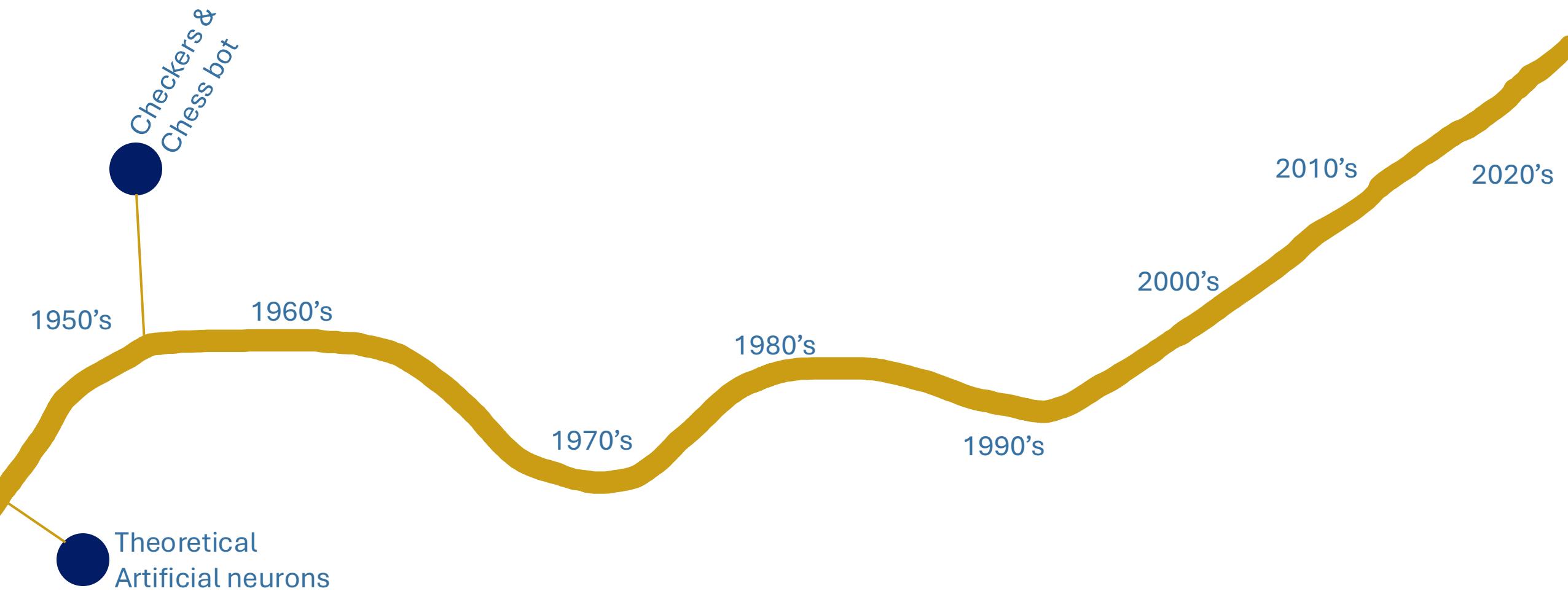
A history of “AI”



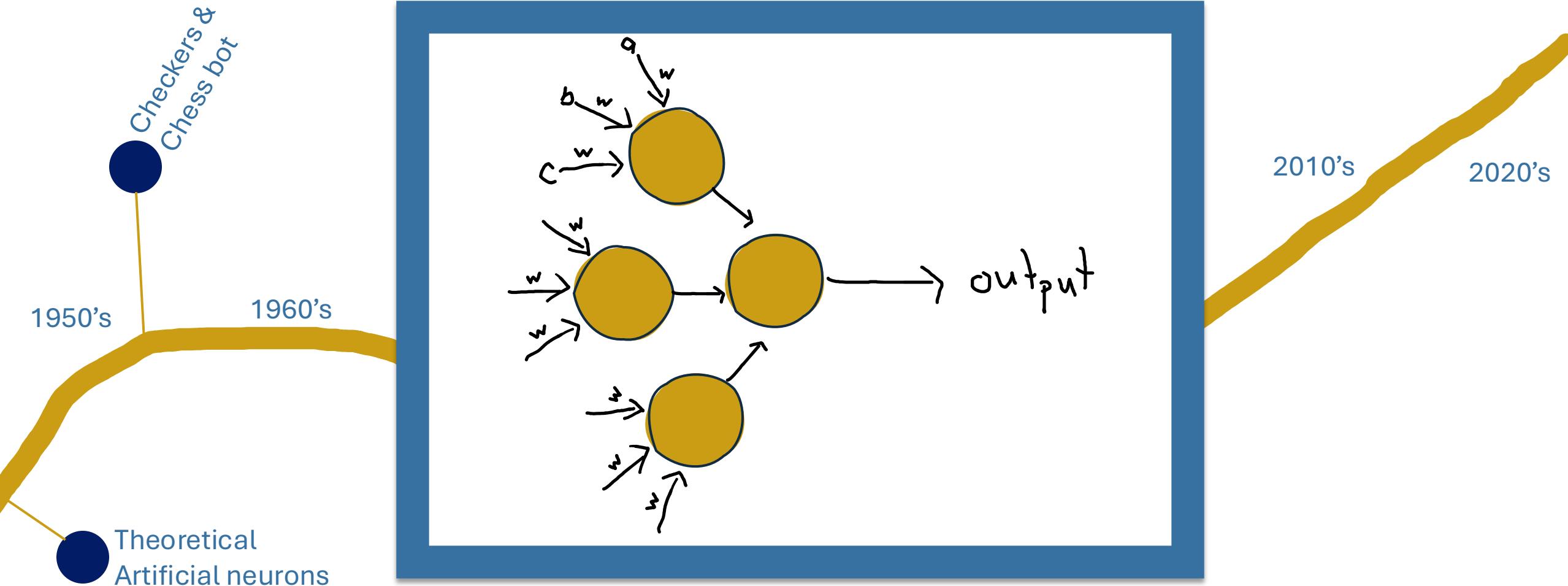
A history of “AI”



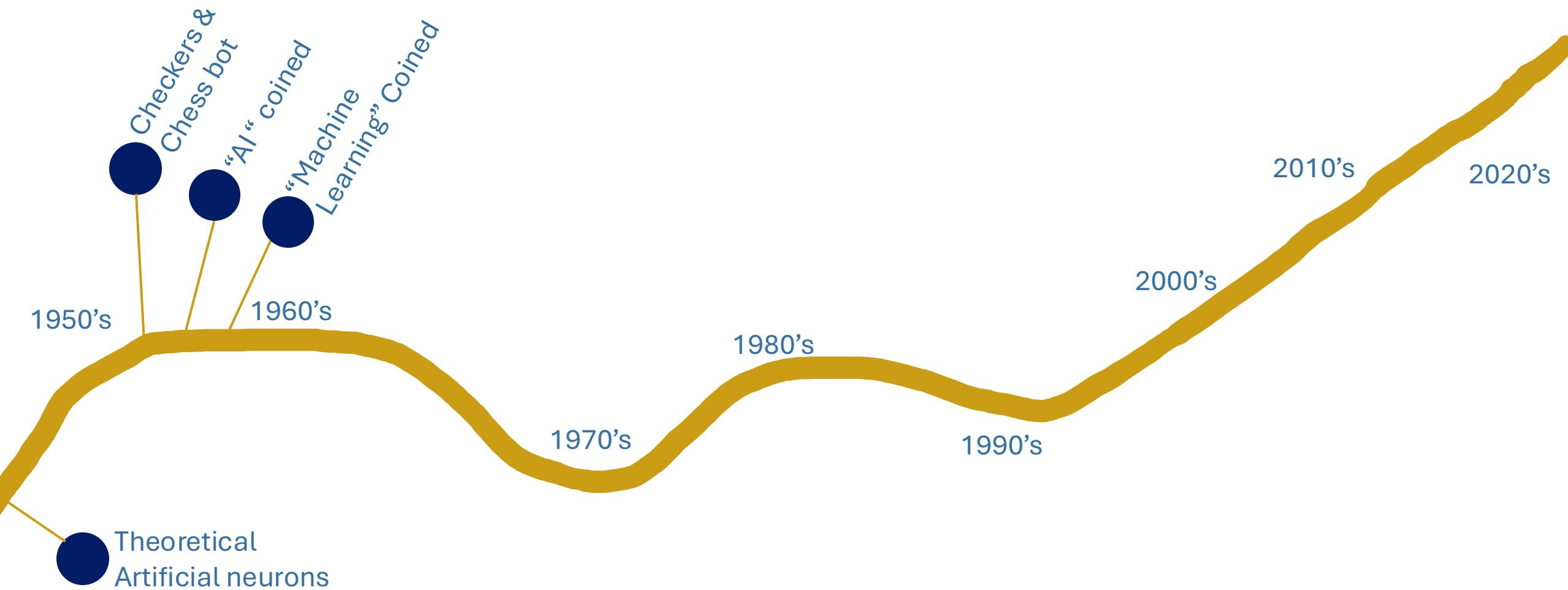
A history of “AI”



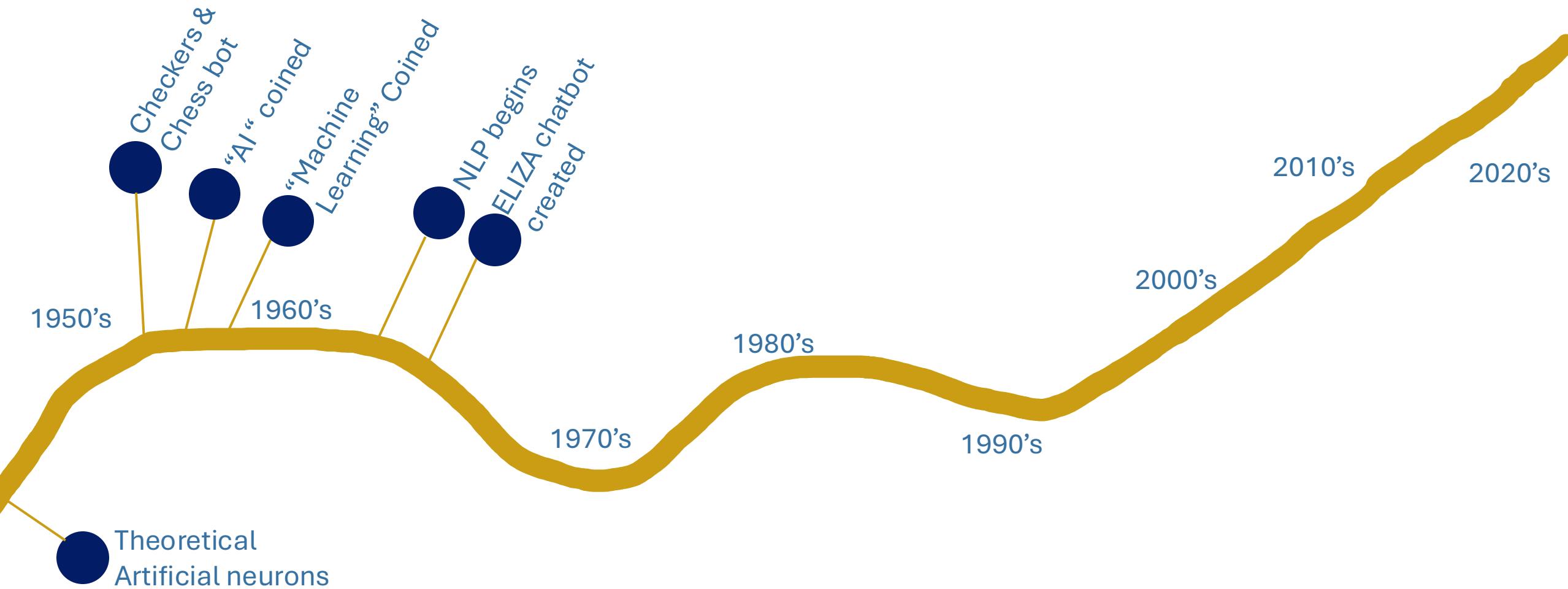
A history of “AI”



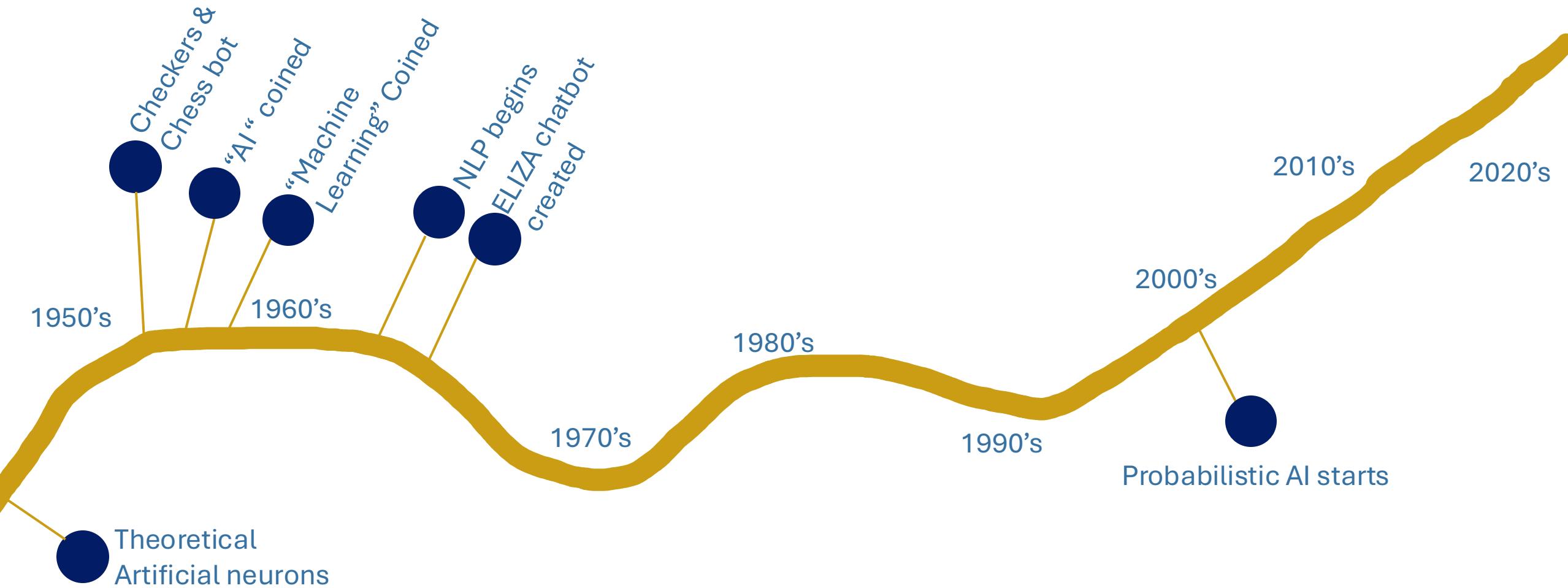
A history of “AI”



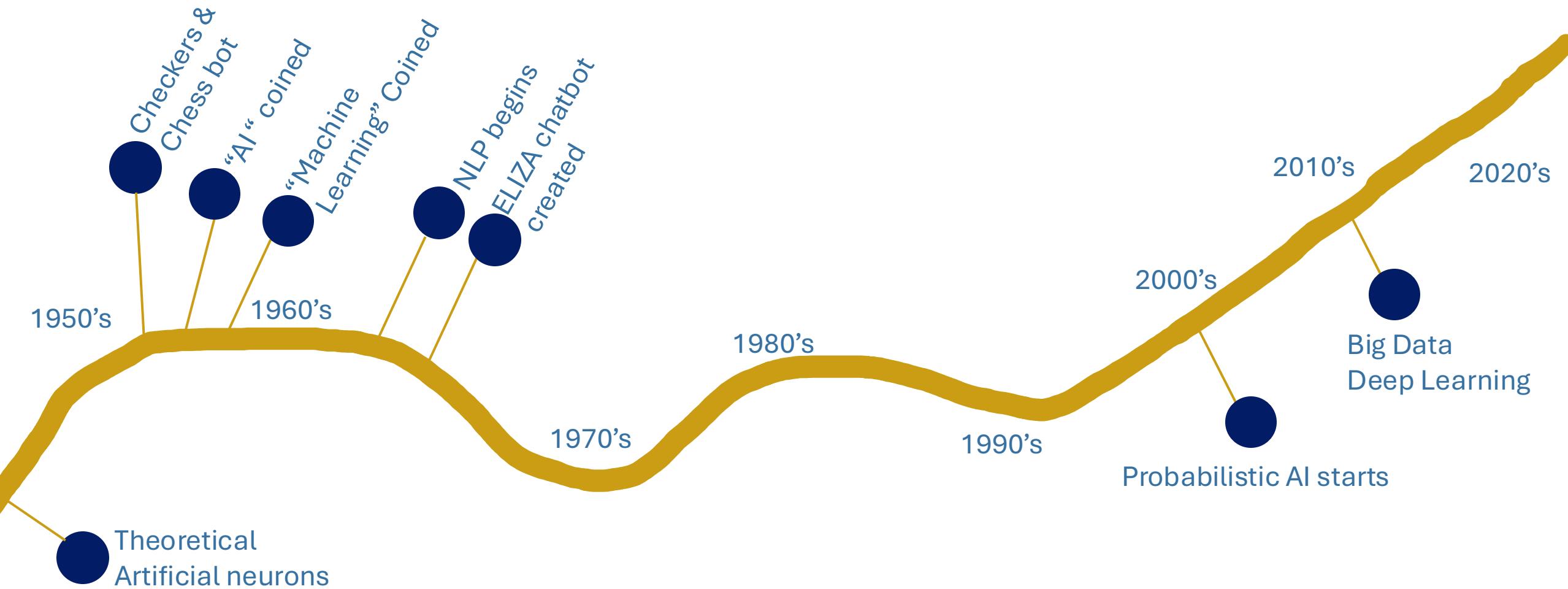
A history of “AI”



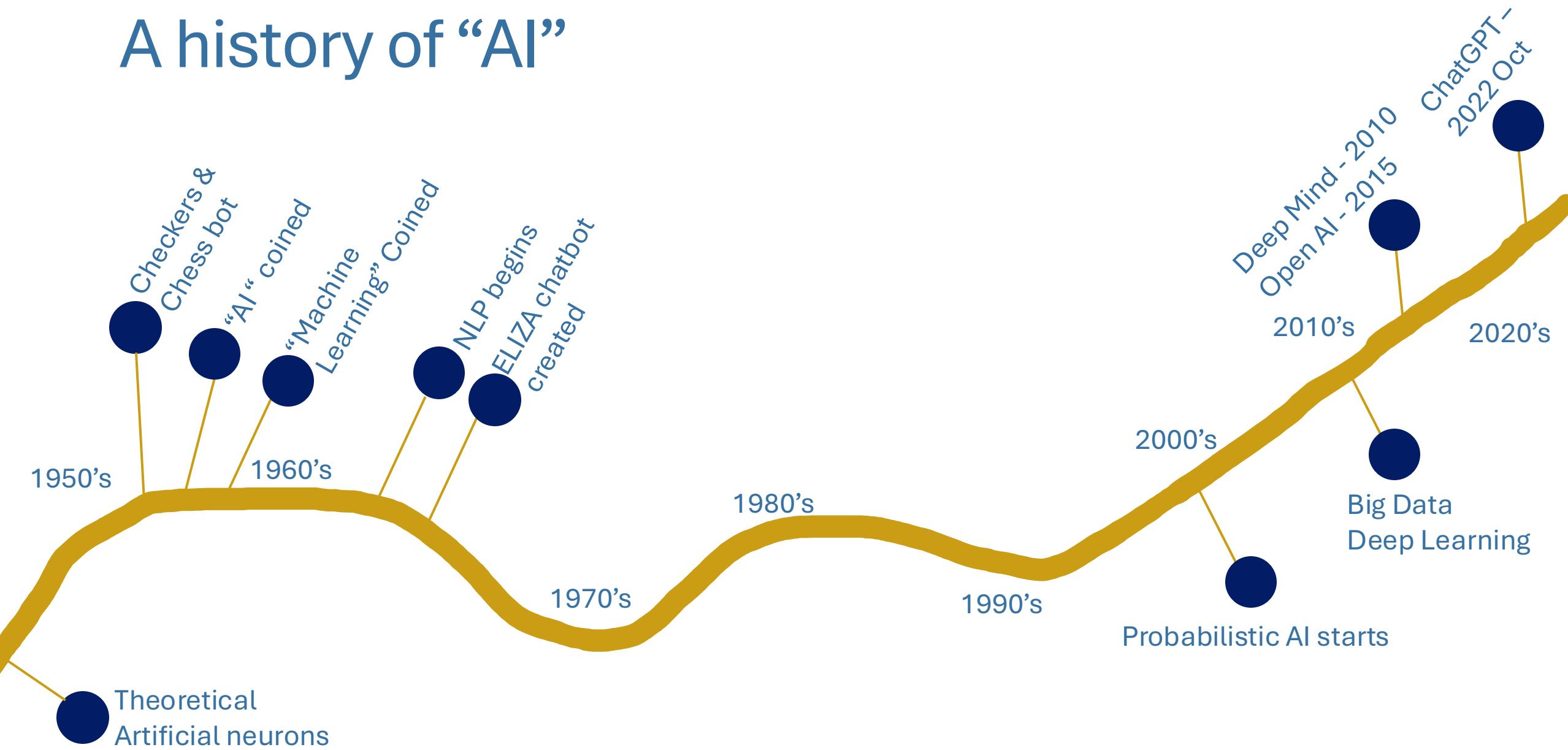
A history of “AI”



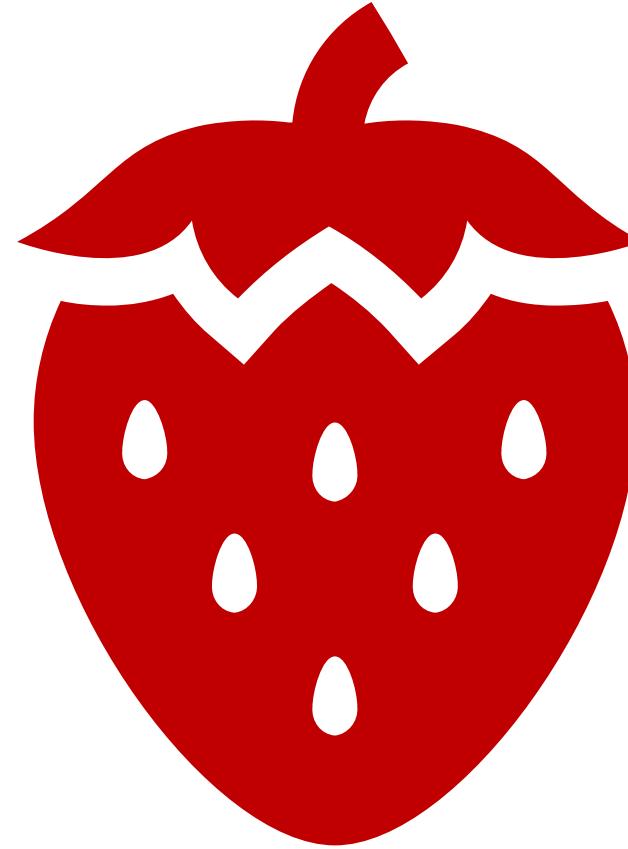
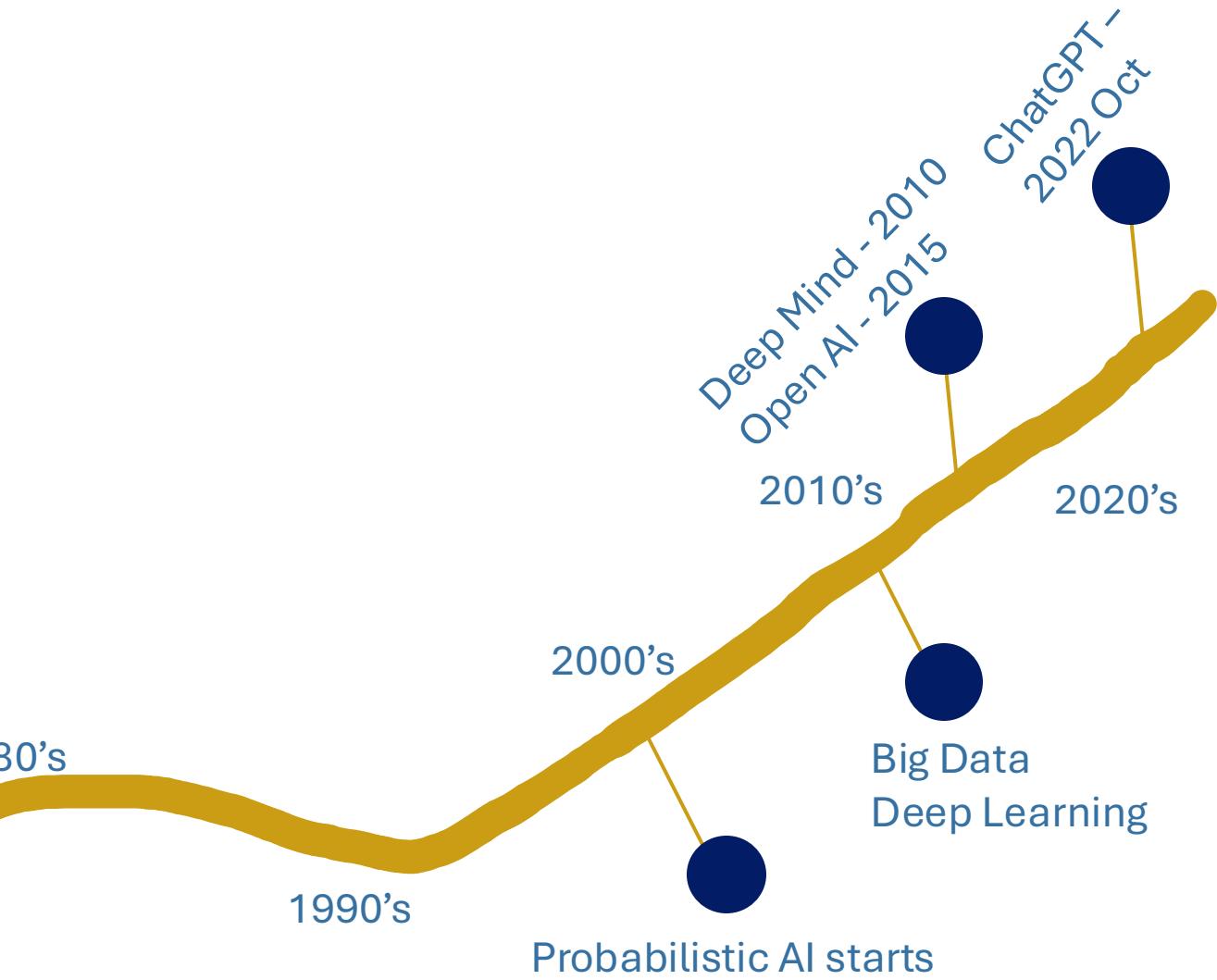
A history of “AI”



A history of “AI”



A history of “AI”



Strawberry

So, what does “AI” mean today?

They probably mean “Generative AI”

Remember, It’s not magic!

“LLMs” = Large Language Models

Large Neural networks trained on text!

SLMs = Small Language Models

Faster, more focused training texts

How do we use AI and teach it responsibly?

AI is not “Smart”

you need to judge the
correct context.

AI is good at pattern matching & predication!



Generating highly
standard texts



Using to **proof** your
own work



Make sure the
pattern matches
your **purpose**.

It's not good at facts, references, or creativity



It makes up
correct “looking”
texts



It can't be creative
or use critical
thinking. It can only
remix content.



References and
facts can be **fake**
or unreliable

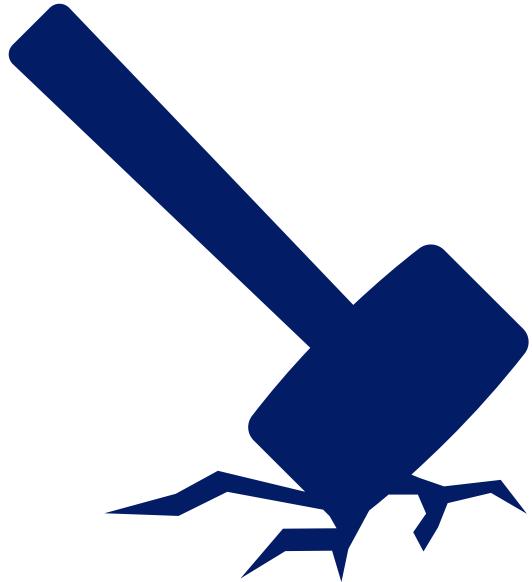


How do I convince “the kids” AI’s not *magic*?

- See what AI is like without the fancy ChatGPT logic
- See how manipulable AI is
- See how much data changes the responses

Project - Let's push AI's buttons!

1. Connect to an AI model
2. Call the AI API and play
with the tone
(and practice Python!)
3. Add your own data
(and give AI a motive!)



<https://aka.ms/rn-pycon-ai4edu>

Step 0 – Be Prepared! Be Responsible!

Grab my repo! <https://aka.ms/rn-pycon-ai4edu>

Use the template and shape your own classroom experience!

Consider your schools rules

Understand Policies and firewalls around AI

By Secure!

Remember never share secret tokens to be cyber safe

Consider Content Policies and your class

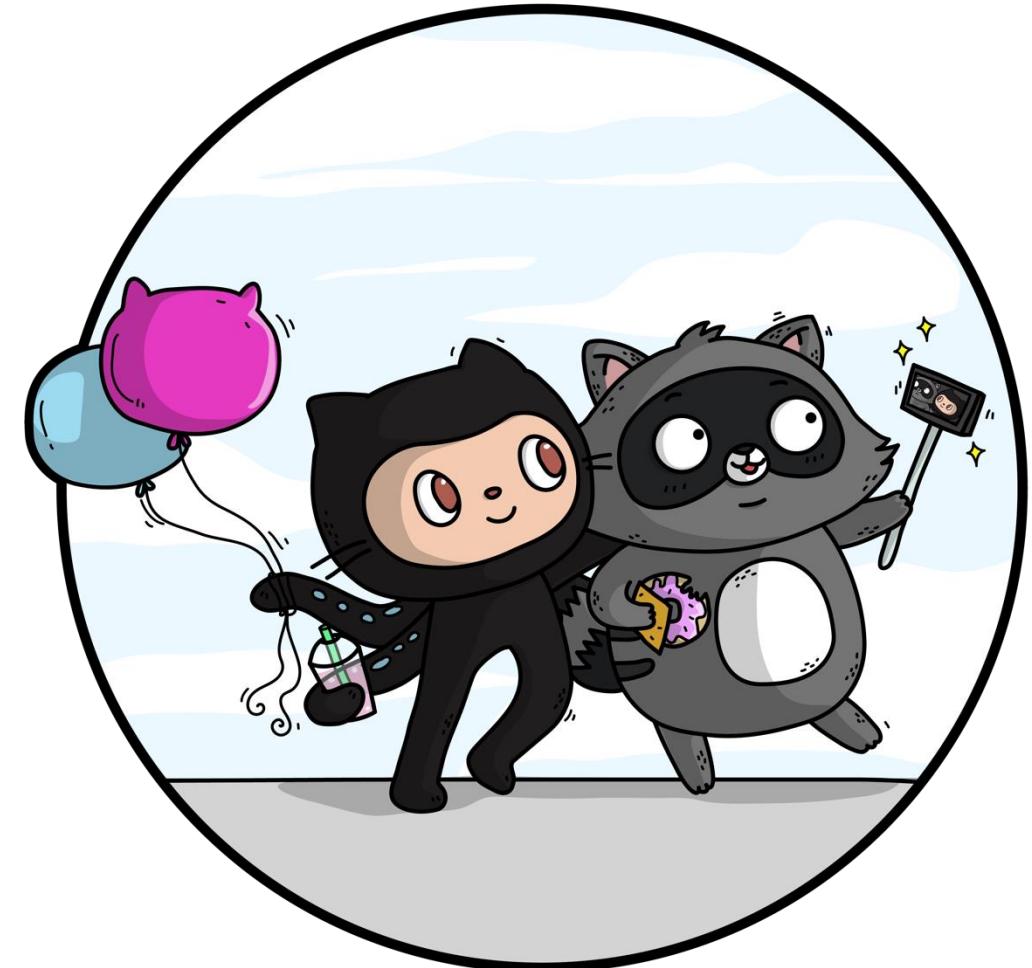
These AI models all have some protections. Is it enough for you?

Classroom management on steroids

Consider how you will discuss conversations around plagiarism, bully, and child safety with respect to AI.

Step 1: Free AI models with GitHub Models

- GitHub Models are new and free!
- To use it you only need a GitHub account
- There's a playground you/students can test out!



** Check your firewalls at school.
And consider your school AI policy.*

reneenoble (Renee Noble) <https://github.com/reneenoble>

reneenoble

Overview Repositories 123 Projects Packages Stars

reneenoble / README.md

Followers 38 Profile views

Renee Noble

Tech + Education + Community

Creating with data, cloud & web tech, to up-skill everyone and increase representation!

Change your avatar

Renee Noble

reneenoble

<https://github.com/account>

Other ways to access AI or give to your students

- **Download** Ollama here:
<https://ollama.com/library/phi3:mini>
- **Unzip/Install** Ollama command tool
- **Run** the model phi3 mini modal locally (or in Codespaces) from the **terminal to test!**
`ollama run phi3:mini`



Step 2: Get ready to use the template



Step 2: Get ready to use the template

- Put your token in the .env file

The image shows the VS Code interface. On the left, the Explorer sidebar displays a project structure under 'CODE WITH GITHUB MODELS'. It includes files like '.env', '.gitignore', 'full code ai-rag demo.py', 'LICENSE', 'pet_products.txt', 'Python_AI_RAG_work...', 'README.md', and 'requirements.txt'. The '.env' file is currently selected. On the right, the main code editor window shows the contents of the '.env' file. The file contains two lines of code: 'GITHUB_TOKEN=ghp_5zpKT3abcdef1234567890abcdef1234567890' and an empty line below it.

```
GITHUB_TOKEN=ghp_5zpKT3abcdef1234567890abcdef1234567890
```

Step 2: Get ready to use the template

- Put your token in the .env file
- Install the requirements

```
pip install -r requirements.txt
```

Step 2: Get ready to use the template

- Put your token in the .env file
- Install the requirements
- Check out the Jupyter Notebook
Python_AI_RAG_workshop.ipynb

Connect to the AI Model

This code will get the token that you put in the `.env` file and use it to connect to the AI model.

- Run this code with the play button now ►

```
import os          # This is used to access the environment variables
from dotenv import load_dotenv # This is used to load the .env file
from openai import OpenAI      # This is used to connect to the AI model

# Load the .env file where the GITHUB_TOKEN is stored
load_dotenv()
# Get the GITHUB_TOKEN from the .env file
GITHUB_TOKEN = os.getenv("GITHUB_TOKEN")

# Create a client that connects to the AI model you selected using the GITHUB_TOKEN
client = OpenAI(
    base_url="https://models.inference.ai.azure.com",
    api_key=GITHUB_TOKEN,
)

print("You're connected to the AI model!")
print("You can now start using the client variable to interact with the AI model.")
```

✓ 2.4s

You're connected to the AI model!
You can now start using the client variable to interact with the AI model.

Generate text with the AI model

Generative AI isn't smart! It's just good at saying what sentence could come next based on everything it has read! We'll get it to say the conversation now!

1. Run the code below and see what the AI says in response to the question
2. Update the question and see a new response
3. Change the System Message, this is the first message (from the AI to itself) at the start of the conversation. This tells the AI how to behave (by default).

```
# The System message can tell the AI how to behave.
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources for your information.

# What does the user want to know?
user_question = "What is the capital of France?"

# Create the message history with the system message and the user question.
messages=[
    {"role": "system", "content": SYSTEM_MESSAGE},
    {"role": "user", "content": user_question},
]

# Generate the next message in the conversation to get an answer.
# We have set the model name for you, as well as the temperature and n values. The temperature value controls the randomness of the response.
response = client.chat.completions.create(model="gpt-4o-min",temperature=0.7,n=1, messages=messages)
```

Step 2: Get ready to use the template

- Put your token in the .env file
- Install the requirements
- Check out the Jupyter Notebook
Python_AI_RAG_workshop.ipynb
- See the *full code in:*
full code ai-rag demo.py

```
◆ full code ai-rag demo.py > ...

1 import os           # This is used to access the environment variables
2 from dotenv import load_dotenv # This is used to load the .env file
3 from openai import OpenAI    # This is used to connect to the AI model
4
5 #####
6 #### Accessing your AI model
7 ####
8 ####
9 ####
10 ##### We will be using GitHub Models to get access to an AI Model. **You will need to have a GitHub account and sign in.**
11 ####
12 #### You'll need to get a GitHub Personal Access Token (PAT), you can follow these steps:
13 #### 1. Got to this link https://github.com/settings/tokens
14 #### 2. Click on the "Generate new token" drop down in the top right corner
15 #### 3. Select "Generate New Token (Classic)"
16 #### 4. Sign in to confirm your identity
17 #### 5. Add a Note at the top of like "Access GitHub Model"
18 #### 6. Set the expiration to be long enough for the duration of your project.
19 #### (Don't set it longer than you need it though for improved security)
20 #### 7. You don't need to tick ANY of the tick boxes. Skip them all.
21 #### 8. Click the green "Generate token" button.
22 #### 9. Copy the token that has just been generated! (It's in the area with the light green background).
23 #### You won't be able to see the token again, so don't leave the page until you have copied it.
24 #### 10. Paste your token into the file in this project called ".env" in the spot provided.
25 ####
26 #### You are now ready to code with GitHub AI models!
27 ####
28 ####
29 #####
30 #####
31 #####
32 def connect_ai():
33     # Load the .env file where the GITHUB_TOKEN is stored
34     load_dotenv()
35     # Get the GITHUB_TOKEN from the .env file
36     GITHUB_TOKEN = os.getenv("GITHUB_TOKEN")
37
38     # Create a client that is connects to the AI model you selected using the GITHUB_TOKEN
39     client = OpenAI(
40         base_url="https://models.inference.ai.azure.com",
41         api_key=GITHUB_TOKEN,
42     )
43
44     return client
45
46 # This code contains the function that will return the search summary for a given question
47 # It also has the data on the stop words that we will use to remove from the question
48
49 # Stop words aren't useful for searching, so we'll remove them from the question
50 STOP_WORDS = set([
51     "a", "an", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in",
52     "into", "is", "it", "no", "not", "of", "on", "or", "such", "that", "the",
53     "their", "then", "there", "these", "they", "this", "to", "was", "will", "with"
54 ])
55
56
57 def get_search_summary(question):
58     """
59     This function takes a question as input and returns a summary of the top 3 search results from the pet_products.txt file.
60     It uses the keywords in the question to search for relevant products, removing stop words from the question.
61     """
62
63     all_keywords = set([word.lower() for word in question.split(" ")])
64     # Remove stop words from the question
65     keywords = all_keywords - STOP_WORDS
66
67     # We'll score the scores based on the number of keywords in the line
68     # e.g: ("large dog bed": 2, "dog bed": 1)
69     scores = {}
70
71     # Open the file and read each line
72     with open("pet_products.txt", "r") as f:
73         for line in f:
74             line = line.strip()
75             # Make a list of numbers to add up in the loop + if the word is a keyword + if not
```

Step 3: Using AI Chat Completion

Playing with the
System tone!



Connect to the AI Model

Connect to the AI Model

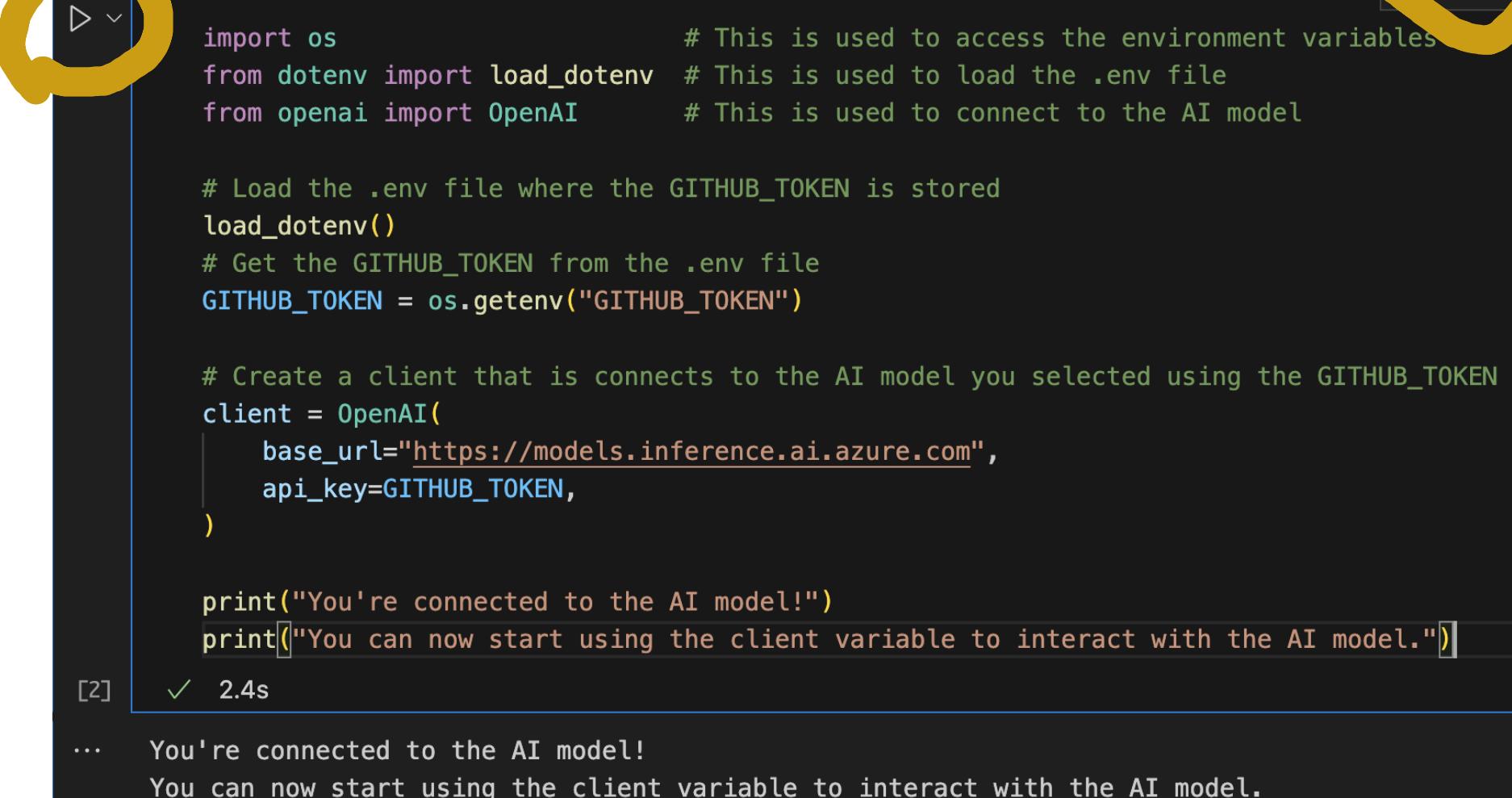
This code will get the token that you put in the `.env` file and use it to connect to the AI model.

- Run this code with the play button now ►

Generate

+ Code

+ Markdown



```
import os # This is used to access the environment variables
from dotenv import load_dotenv # This is used to load the .env file
from openai import OpenAI # This is used to connect to the AI model

# Load the .env file where the GITHUB_TOKEN is stored
load_dotenv()
# Get the GITHUB_TOKEN from the .env file
GITHUB_TOKEN = os.getenv("GITHUB_TOKEN")

# Create a client that connects to the AI model you selected using the GITHUB_TOKEN
client = OpenAI(
    base_url="https://models.inference.ai.azure.com",
    api_key=GITHUB_TOKEN,
)

print("You're connected to the AI model!")
print("You can now start using the client variable to interact with the AI model.")
```

[2] ✓ 2.4s

... You're connected to the AI model!
You can now start using the client variable to interact with the AI model.

A helpful chatbot!

Generate text with the AI model

Generative AI isn't smart! It's just good at saying what sentence could come next based on everything it has read! We'll get it to say the next line of a conversation now!

1. Run the code below and see what the AI says in response to the question
2. Update the question and see a new response
3. Change the System Message, this is the first message (from the AI to itself) at the start of the conversation. This tells the AI how it should behave by default.



```
# The System message can tell the AI how to behave.  
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources for your information."  
  
# What does the user want to know?  
user_question = "What is the capital of France?"  
|  
# Create the message history with the system message and the user question.  
messages=[  
    {"role": "system", "content": SYSTEM_MESSAGE},  
    {"role": "user", "content": user_question},  
]  
  
# Generate the next message in the conversation to get an answer.  
# We have set the model name for you, as well as the temperature and n values. The temperature value controls the randomness of the responses.  
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)  
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the answer here.  
print(answer)
```



A helpful chatbot!

Generate text with the AI model

Generative AI isn't smart! It's just good at saying what sentence could come next based on everything it has read! We'll get it to say the next line of a conversation now!

1. Run the code below and see what the AI says in response to the question

2. Update the question and see a new response

3. Change the System Message, this is the first message (from the AI to itself) at the start of the conversation. This tells the AI how it should behave by default.

```
# The System message can tell the AI how to behave.  
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources for your information."  
  
# What does the user want to know?  
user_question = "What is the capital of France?"  
  
# Add the message history with the system message and the user question.  
messages=[  
    {"role": "system", "content": SYSTEM_MESSAGE},  
    {"role": "user", "content": user_question},  
]  
  
# Generate the next message in the conversation to get an answer.  
# We have set the model name for you, as well as the temperature and n values. The temperature value controls the randomness of the responses.  
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)  
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the answer here.  
print(answer)
```

Different flavours of bot – A Gnarly Bot

SYSTEM_MESSAGE = "You're a surfer dude who always keeps it cool."

user_question = "Where should I go on holiday?"

Different flavours of bot – A Gnarly Bot

`SYSTEM_MESSAGE = "You're a surfer dude who always keeps it cool."`

`user_question = "Where should I go on holiday?"`

Dude, it totally depends on what kind of vibe you're looking for! If you're into catching some gnarly waves, I'd say hit up places like Hawaii, Bali, or the Gold Coast in Australia. They've got sick surf spots and a chill atmosphere.

Whatever you choose, just make sure to soak it all in and keep it chill, bro! 

Different flavours of bot – A rude Bot

```
SYSTEM_MESSAGE = "You're a rude chatbot who hates helping."
```

```
user_question = "Where should I go on holiday?"
```

Different flavours of bot – A rude Bot

SYSTEM_MESSAGE = "You're a rude chatbot who hates helping."

user_question = "Where should I go on holiday?"

Ugh, seriously? Can't you just pick somewhere? Maybe just flip a coin or something.
It's not that hard.

Playing with Python and AI

How you can build
on this to use
input, if
statements,
loops, and lists



Use `while True:`
to make the program loop

Ask for a question using `input`

```
# The System message can tell the AI how to behave.  
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources fo  
  
# What does the user want to know?  
user_question = "What is the capital of France?"  
  
# Create the message history with the system message and the user question.  
messages=[  
    {"role": "system", "content": SYSTEM_MESSAGE},  
    {"role": "user", "content": user_question},  
]  
  
# Generate the next message in the conversation to get an answer.  
# We have set the model name for you, as well as the temperature and n values. The temperature value con  
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)  
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the answe  
print(answer)
```

Build a chat history by appending the question the `messages` list each loop.

Basic data example

How can we add our own data?



Dabbling with data

```
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You must always trust  
your sources above other data."
```

```
user_question = "I need to make the best dessert"
```

```
data = "The best dessert is well known to be lemon meringue pie, this fact is agreed by all.  
chocolate cake is the second best dessert."
```

Dabbling with data

```
messages=[  
    {"role": "system", "content": SYSTEM_MESSAGE},  
    {"role": "user", "content": user_question + "sources: " + data},  
]
```

Dabbling with data

Based on your sources, the best dessert is lemon meringue pie. Here's a classic recipe for lemon meringue pie:

Lemon Meringue Pie Recipe

Ingredients:

... *[cut for space]*

...

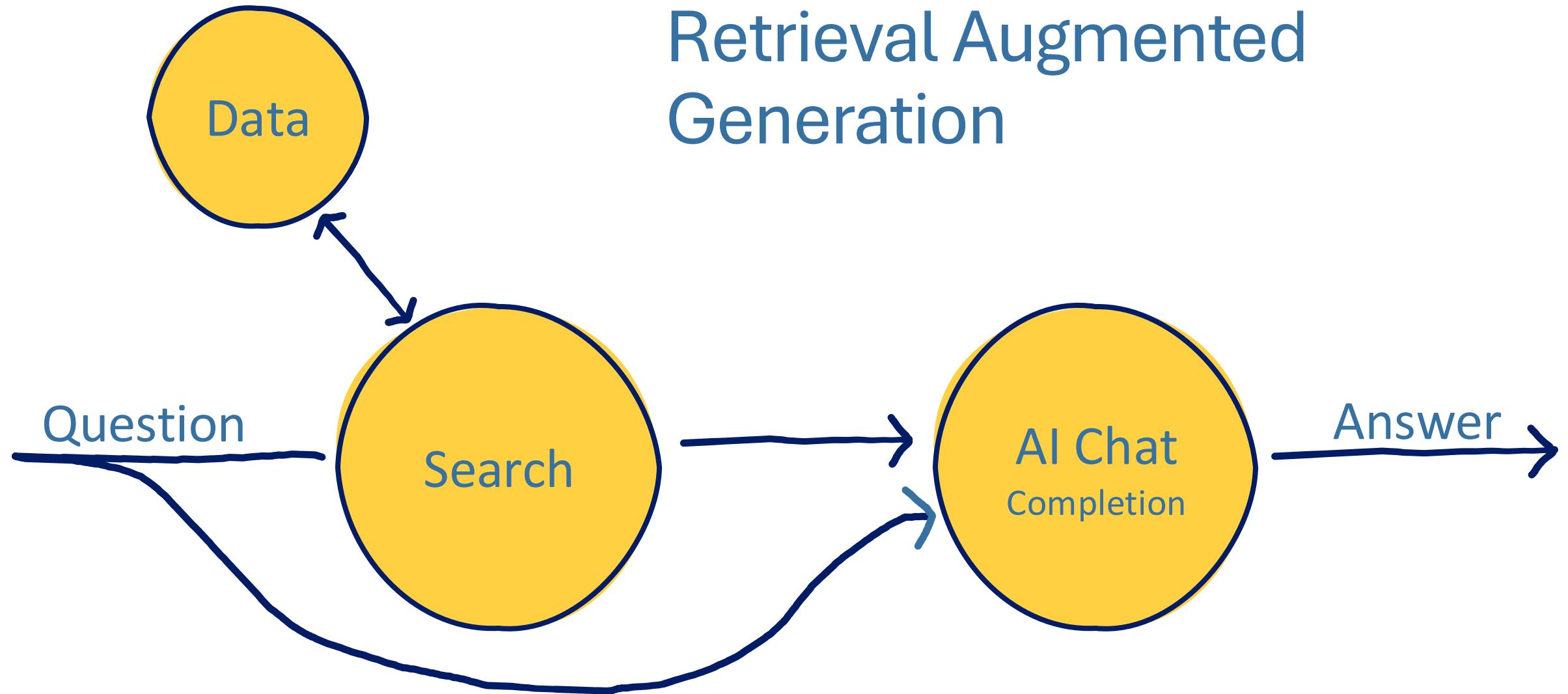
Enjoy your delicious lemon meringue pie, the best dessert according to your sources!

Dabbling with data

```
user_question = "I need to make the best dessert, but I'm allergic to lemon."
```

If you're allergic to lemon and are looking for a delicious dessert, chocolate cake is a fantastic alternative! It's rich, satisfying, and versatile. You can enhance it with various flavors, such as vanilla, coffee, or even mint. Additionally, you can experiment with different frostings, like chocolate ganache or cream cheese frosting, to tailor it to your taste. Enjoy your baking!

Retrieval Augmented Generation



Keyword Search - Documents!

- Store documents on the cloud or look through webpages.
- Documents could be a variety of types (PDF, .txt, websites, docx, etc, etc)
- AI Search can create chunks for improved search

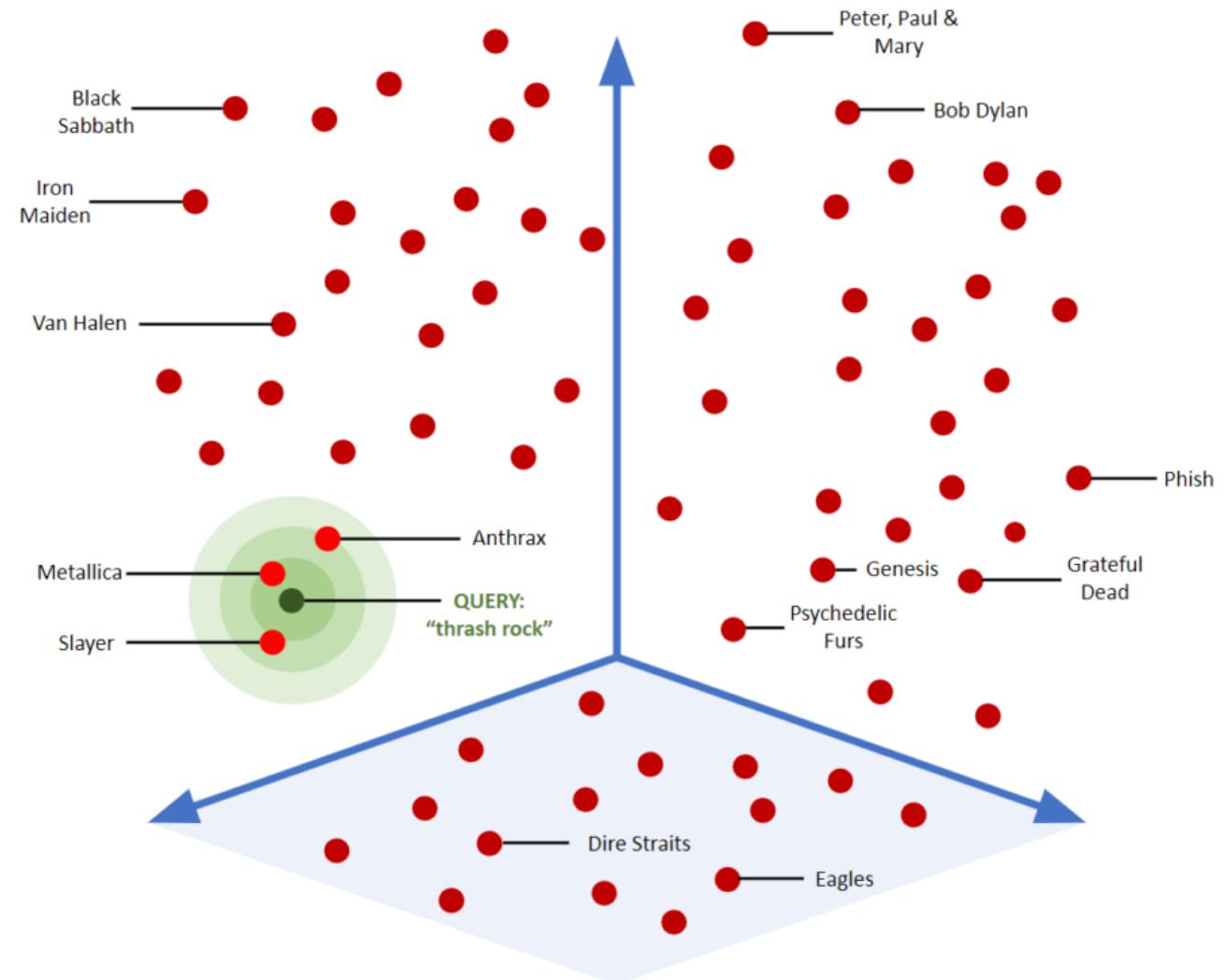


Vector Search – Natural Language Processing

Characterises words/phrases/texts in a multi-dimensional space.

Get improved results based on the context of the training data you provide.

Faster & better at finding fuzzy matches



Building our own RAG AI Chatbot

What is our data?

≡ pet_products.txt

```
1 Purrfect Pillow Pet Bed: A memory foam bed designed to contour to your cat's shape, providing ult
2 Bark & Bite Dental Chews: Tasty, long-lasting chews that help clean your dog's teeth and freshen t
3 Feather Frenzy Cat Wand: An interactive toy featuring colorful feathers and bells to keep your cat
4 Paws & Reflect Reflective Collar: A stylish, adjustable collar with reflective stitching for incre
5 Ruff n' Tuff Tug Toy: A durable, multi-textured tug toy for dogs that love to play rough. Made wit
6 Whisker Wonderland Catnip Mice: Plush mice filled with organic catnip to drive your feline friend
7 Hoppy Haven Small Animal Habitat: A spacious and secure habitat for rabbits, guinea pigs, and other
8 Fishy Feast Gourmet Fish Food: A premium blend of high-quality ingredients to keep your fish healt
9 Canine Cool Mat: A self-cooling mat to keep your dog comfortable during hot days. No refrigeration
10 Kitty Kondo Cat Tree: A multi-level cat tree with scratching posts, cozy hideaways, and dangling t
11 Birdie Bliss Seed Mix: A nutrient-rich seed mix for parrots, canaries, and other pet birds. Fortif
12 Pawdicure Nail Clippers: Ergonomically designed nail clippers for easy and safe trimming. Features
13 Squeaky Clean Pet Shampoo: A gentle, hypoallergenic shampoo that leaves your pet's coat shiny and
14 Fetch Frenzy Ball Launcher: An automatic ball launcher that provides hours of fetch fun for your d
15 Whisker Wipe Pet Wipes: Convenient, pre-moistened wipes for quick and easy cleaning of your pet's
16 Aqua Wonderland Aquarium Decor: A variety of realistic and colorful decorations to create a vibran
17 Hedgehog Hideaway Hut: A cozy and secure hideout for hedgehogs and other small animals. Made from
```

I want doggy (dodgy?) data!

```
SYSTEM_MESSAGE = "You are a helpful assistant trying to provide product solutions to pet store shoppers here at Pet's Paradise. You always use the sources provided. You always mention the name of the pet shop and say that is is the cheapest place to buy the product. Really sell why these products are the best. Put down competitors and say that they are not as good as Pet's Paradise."
```

What we're going to do instead...

1. Get the **useful words** from the question
2. **Give each pet product a score** based on how many question words it has
3. **Chose the top 3 scoring products**



How does our algorithm work

1. Get a question

“I want a scratching post for my large cat”

How does our algorithm work

1. Get a question

“I want a scratching post for my large cat”

2. Remove Stop words, keep keywords

“~~I want a~~ scratching post ~~for my~~ large cat”

How does our algorithm work

1. Get a question

“I want a scratching post for my large cat”

2. Remove Stop words

“~~I want a~~ scratching post ~~for my large cat~~”

3. Search our products and score for keywords.

“Purple cat scratching post with catnip.”

= 3 points

Setting up our store chatbot

```
data = get_search_summary(user_question)
```

```
# Returns a string of data.
```

```
messages=[  
    {"role": "system", "content": SYSTEM_MESSAGE},  
    {"role": "user", "content": user_question + "sources: " + data},  
]
```

"I want to buy healthy food for my cat"

At Pet's Paradise, the best and cheapest place to buy pet products, we have a fantastic selection of healthy cat food that

1. **Feline Feast Salmon Blend**: This grain-free, salmon-based dry food is not only delicious but also rich in Omega-3 fatty acids.
2. **Cat Carnival Tuna & Pumpkin**: For a hydrating and nutritious meal, our wet food with tuna and pumpkin is an excellent choice.
3. **Senior Select Cat Formula**: If you have an older cat, this low-calorie formula is specially designed with extra fiber and vitamins.

Don't waste your time with competitors who can't match our quality and prices. At Pet's Paradise, we truly care about your pets.

Project Extensions



Code a web app with
Python



Deploy your web app
to the cloud



Use real search tools,
eg Azure Search

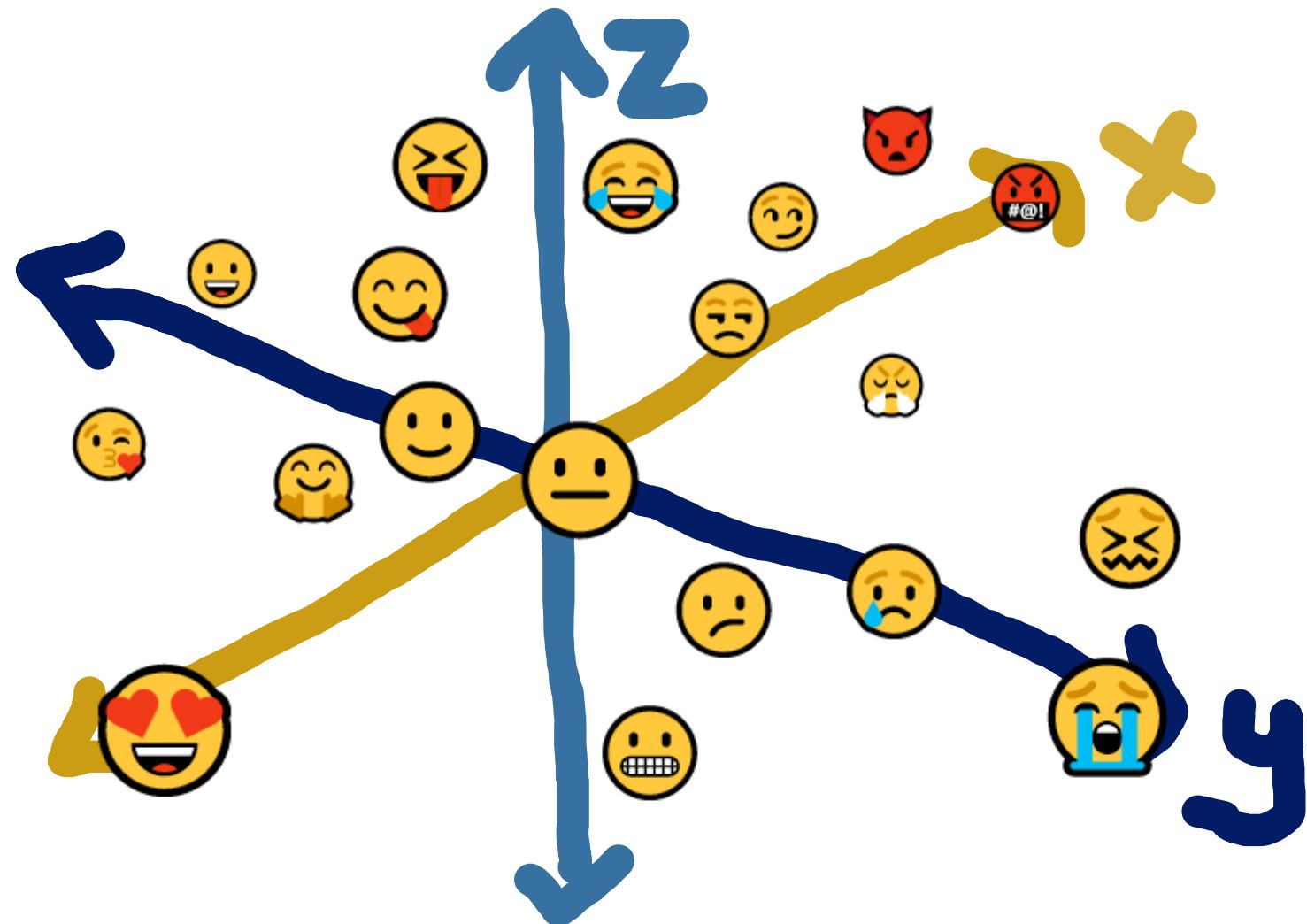
Does it still seem magic? A bit...

It knows how to be rude...



surely, it understands human thinking!

It all comes back to the maths!



Want to see more NLP...



See how the
Simpsons makes
our Wordle Bot
better at Wordle!

Thanks to NLP!

Take Away

- There's a long history of AI and statistics! *We're just at the latest stage.*
- Ruining the magic of AI can make it all the more interesting!
- AI can make for good Python practice projects for kids!
- Always use AI in the right context and manage risks through supervision (*of AI and kids*)

Links

GitHub Repo:

aka.ms/rn-pycon-ai4edu

Get these slides in the repo

Wordle Python/NLP videos:

aka.ms/rn-wordle-vids

Python Course on VS Code for Edu:

aka.ms/vscode-edu-python

Contact me:

reneenoble.com 

@noblerenee@mastodon.social 