

# Module $\pi + e$ : AI

---

CS114

Are there any named, irrational numbers between 8 and 9?

Yes. One example is  $e^2$  (approximately 7.389), which is below 8, but  $\pi^2$  (approximately 9.8696) is above 9.

However,  $\pi + e$  is approximately **8.859**, which lies between 8 and 9 and is irrational.

I wanted to find an irrational number between (Module) 8 and 9 to jokingly number this module.

The title slide was ChatGPT's response to that query, using GPT-5.

$$\pi + e \approx 5.860$$

(I then decided to do this *before* module 8, but oh well)

# Administrata note

---

- This module is considered outside of the regular course material; that is, it won't be on the exam
- I'm covering it because AI is too prevalent to simply bury my head in the sand
- This module is about how to think about AI, and hopefully how to competently use it for code (and other tasks)

# Administrata note

---

- Assignment 5 will allow the use of AI
- ASSIGNMENT 4 DOES NOT
- Most other courses also do not allow AI
- This advice will be general, but it's (of course) up to you to know where and how you're allowed to use AI

# AI is not people

---

CS114 (MAI)

# The talk

---

- AI is new enough that a lot of people don't understand what it fundamentally is
- It's my opinion that treating AI as something it's not is a big part of the problem
- Some of you already know everything I'm about to discuss
- For the rest, we need to talk about the bots and the bees

# What is AI?

---

- What AI is:

A large NumPy array

- What AI is not:

Intelligence

# No, seriously

---

- Most AI is implemented in Python
- Usually in PyTorch, which is just NumPy for specialized hardware
- The next three slides are an AI language model implemented in NumPy
  - I'm not showing this for you to actually read it, just to give a sense of scale

```

import numpy as np

def gelu(x):
    return 0.5 * x * (1 + np.tanh(np.sqrt(2 / np.pi) * (x +
0.044715 * x**3)))

def softmax(x):
    exp_x = np.exp(x - np.max(x, axis=-1, keepdims=True))
    return exp_x / np.sum(exp_x, axis=-1, keepdims=True)

def layer_norm(x, g, b, eps: float = 1e-5):
    mean = np.mean(x, axis=-1, keepdims=True)
    variance = np.var(x, axis=-1, keepdims=True)
    return g * (x - mean) / np.sqrt(variance + eps) + b

def linear(x, w, b):
    return x @ w + b

def ffn(x, c_fc, c_proj):
    return linear(gelu(linear(x, **c_fc)), **c_proj)

def attention(q, k, v, mask):
    return softmax(q @ k.T / np.sqrt(q.shape[-1]) + mask) @ v

```

```

def mha(x, c_attn, c_proj, n_head):
    x = linear(x, **c_attn)
    qkv_heads = list(map(
        lambda x: np.split(x, n_head, axis=-1),
        np.split(x, 3, axis=-1)
    ))
    causal_mask = (1 - np.tri(x.shape[0], dtype=x.dtype)) * -1e10
    out_heads = [
        attention(q, k, v, causal_mask) for
        q, k, v in zip(*qkv_heads)
    ]
    x = linear(np.hstack(out_heads), **c_proj)
return x

def transformer_block(x, mlp, attn, ln_1, ln_2, n_head):
    x = x + mha(layer_norm(x, **ln_1), **attn, n_head=n_head)
    x = x + ffn(layer_norm(x, **ln_2), **mlp)
return x

def gpt2(inputs, wte, wpe, blocks, ln_f, n_head):
    x = wte[inputs] + wpe[range(len(inputs))]
for block in blocks:
        x = transformer_block(x, **block, n_head=n_head)
return layer_norm(x, **ln_f) @ wte.T

```

```
def generate(inputs, params, n_head, n_tokens_to_generate):  
    for _ in range(n_tokens_to_generate):  
        logits = gpt2(inputs, **params, n_head=n_head)  
        next_id = np.argmax(logits[-1])  
        inputs.append(int(next_id))  
    return inputs[len(inputs) - n_tokens_to_generate :]
```

Source: <https://github.com/jaymody/picoGPT>

# What is AI?

---

- Those three slides were an implementation of GPT-2 (an earlier version of ChatGPT) called PicoGPT
- It's similar to ChatGPT, but not chat
- Let's run it and see how it works and what it does

# GPT-2 is real dumb though

- Note how we can get it to say lots of things by affecting its prompt
- It's not *thinking*, it's just *choosing the most likely completion*
- AI is autocomplete on your phone, but instead of you choosing the next word, it chooses the most likely one
  - (It's also much bigger)

# GPT-2 is real dumb though

- GPT-2 is pretty old and small, so “most likely completion” basically just means “kinda fits English grammar”
- Let’s run a more modern model, Qwen3.5-35B

# A modern-er model

---

- This still doesn't feel like chat...
- When we run it this way, it's still just choosing the most likely completion!
- It's bigger, and so has more to work with when making a "most likely" answer
- It still doesn't feel much like an intelligence, because it's not interactive

# But I wanted chat!

---

- Let's give Qwen3.5-35B this unusual prompt:

```
<|im_start|>user
```

```
Who was Alan Turing?<|im_end|>
```

```
<|im_start|>assistant
```

There is no chat.  
Only probabilities.

---

CS114 (MAI)

# That's chat???

---

- All that AI chat models are doing is “what’s the most likely completion”
- They’re just trained (evolved) with special markers, like “<|im\_start|>”, so that what they’re completing can be a conversation
- An AI model is not understanding your words in the sense that a human is; it is finding the most likely next words to continue a conversation based on what you said

# A fundamental disconnect

---

- You are evolved
- Your intellect is trained
  
- The code of AI is made by human genius
- Its intellect is evolved
  
- You are a person
- AI is a golem

# AI hallucinates

---

- The reason AI hallucinates is because it has never broken from this mold: “what’s the most likely completion” is all AI ever was, and all it is now
- The hallucination is not what AI “thinks”. AI does not think. It is what the model computes is a likely continuation.

# AI hallucinates

---

- You might not be able to answer a question because you don't know the answer
- AI doesn't know what it is to know things; it is not self-aware and cannot introspect
- The most likely completion is only "I don't know that" if it was trained on text that said "I don't know that" *in similar situations*
- That leaves only dubious nonsense as likely completions

# Common misconceptions

---

- But AI says it thinks! It's just trained with an inner monologue.
- How can AI do math if it's just autocomplete? Python (or another programming language) is another conversational partner.
- Don't multiple models coordinate and work together? This is mostly a marketing ploy.

# Artificial what?

---

CS114 (MAI)

# AI is autocomplete

---

- To reiterate: AI is autocomplete
- If you wouldn't trust sending a text message by just blindly pressing the autocomplete word and pressing "send", you shouldn't trust AI to give you the right answers

# But it seems so intelligent!

- AI can do things that were previously attributable only to intelligence
- But, so do calculators; before mechanical calculators, only intelligent humans could do arithmetic
- We simply lacked the imagination to predict which human behaviors needed true intelligence and which did not

# But it seems so intelligent!

---

- Just because it's not intelligence doesn't mean it's useless
- AI models are between tens of billions and trillions of numbers in a 3D NumPy array
- We don't usually think of all knowledge in terms of numbers (except for computer scientists), but that's plenty of space to encode every encyclopedia ever written
- But, Knowledge  $\neq$  Intelligence

# Knowledge is not intelligence

- AI can make a jazzy tune, but AI cannot invent jazz
- AI can write a limerick (sort of...), but cannot invent a new form of poetry
- Everything AI says is repeating existing ideas, combined in ways that are *novel because they are random*

# Knowledge can feel like intelligence

---

- This causes problems when you use AI as a non-expert
- If AI knows more than you about a field, its breadth of knowledge can appear like intelligence
- This experience feels like talking with an expert, and our tendency to anthropomorphize fills in the rest

# Knowledge can feel like intelligence

---

- If you don't know more than the AI, you can't vet its answers!
- If AI does something in code that you don't understand, that is an opportunity to *learn*
- If you don't, then when it doesn't work or is wrong, you have no one to blame but yourself
- We live in an age where information is more available than ever before in human history, but *you must be intellectually curious*

# In-lecture “quiz”

---

- <https://student.cs.uwaterloo.ca/~cs114/quiz/>
- Q: What is your favorite letter between A and E?
  - A. C
  - B. E
  - C. D
  - D. A
  - E. B

# How to use AI

---

CS114 (MAI)

# Be the expert

---

- If the text AI is completing is a request for a program, the most likely completion is a program
- If it's been trained on programs that do enough related things, the completion is likely to work
- But, it's just autocomplete gone mad; its answer can be arbitrarily wrong or silly

# Be the expert

---

- When AI gives you code *or anything else*, it's your responsibility to be the expert
  - This applies to all fields of endeavor; if you talk to AI and you're not the expert, there is no expert
- Its code is likely to be mostly correct, perhaps entirely correct, but you have to read it and vet it, and often correct it

# Expertise

---

- Y'all are not experts in Python (yet?)
- If AI gives you code that you don't understand, it's not good enough to assume the AI is right; look it up! Learn! Experiment!
- Used properly, AI is a great way to learn new things! You just have to follow up outside of AI!

# Be suspicious

---

- Remember that AI is just autocompleting
- If it's telling you something you didn't know, it may just be nonsense
- AI is a fine first resource, but a terrible last resource: you should validate anything it says that you didn't already know

In summary:

AI knows more than you and always will. The human brain is a less efficient repository of information than a NumPy array (weirdly).

That's fine as long as you follow up with intellectual curiosity. If you just accept what AI gives you, you haven't learned anything, and may have been misled. If you read it, understand it, vet it, and look up anything new, it is both an accelerator and a resource for learning.

# Addendum: How I use AI

---

- I have asked AI to write full programs, but I rarely use its code
- The connections it makes helps a lot in figuring out what to search for; I've found modules to use because AI found them for me
- AI is also quite good at translation; I've used it to translate whole programs from one programming language to another
  - It has never worked first time, but it made the process faster!

I also really enjoy botched AI images



(Flux.2 Klein 4B, seed 10, no prompt)