BIOL 2209L: AI and energy

Ken Aho

2025-09-18

# 1 AI and LLMs

The term *Artificial intelligence* (AI) refers to the capacity of a computer to perform tasks typically associated with human intelligence (Wikipedia 2025a). The field of AI has several non-mutually-exclusive foci. These include:

* *Machine learning*, part of AI research since its beginning in 1956, machine leaallows a program to improve its performance on a given task automatically (Russell and Norvig 2021).
* *Machine perception* is the capacity of a computational device to receive and use input from sensors (Russell and Norvig 2021). Examples include software for speech recognition, image classification, and facial recognition.
* *Natural language processing* (NLP) describes the capacity of a computer program to read and communicate in human languages.

A number of recent developments, including *transformers* –a neural network framework that weighs the importance of text components and creates responses through a word embedding table (Wikipedia 2025d)– and *Large Language Models* (LLMs) –machine learning entities trained on a vast amount of text, for the purpose of language processing and language generation (Wikipedia 2025c)– have resulted in dramatic improvements in NLP performance. The most sophisticated LLMs are *Generative Pre-trained Transformer* (GPT) models. By 2023 GPT models were able to achieve human-level scores on the bar exam and the SAT test, among other applications (Bushwick 2023).

GPT models may generate biased reports –largely as a result of biased training data (Zack et al. 2024), and hallucinations (plausible-sounding random falsehoods)(Wikipedia 2025b). Despite these drawbacks, however, the capacity of GPT and other AI approaches to emulate (and potentially improve on) processes previously under human proprietorship is clear (and unnerving). Indeed, because of this potential, massive investments (mostly from entrenched software giants, e.g., Microsoft, Oracle, Meta, Google) have been made into AI-associated infrastructure, including the ongoing construction of new data centers and server farms (Van Der Vlist, Helmond, and Ferrari 2024). [Business Insider](https://www.businessinsider.com/us-data-center-construction-40-billion-spend-hits-record-high-2025-9) reported last week that data center construction costs in the US could total more than $1 trillion by 2028 (Fig 1.1).



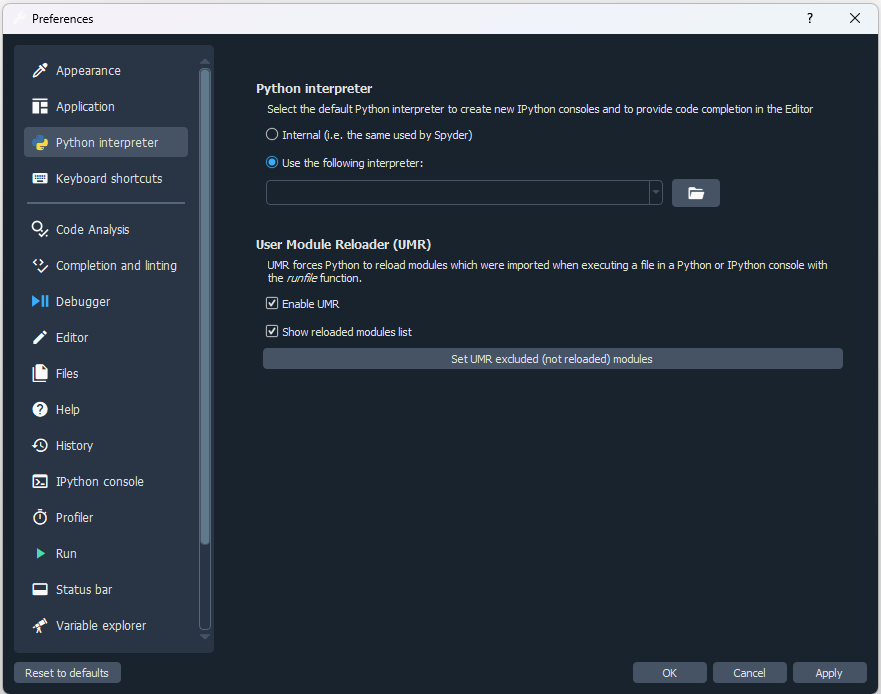
As a result of these efforts, and the high energy use of GPT processes, energy demands of data centers are excepted to double globally by 2030 (Kolbert 2024). (De Vries 2023) estimated that if Google were to integrate generative AI into every search thread, its electricity usage rise to approximately kilowatt-hours per year, exceeding the annual energy demand of many countries, including Kenya, Guatemala, and Croatia.

Today we will briefly explore GPT energy use in a lab using the Python programming language.

# 2 Python

[**Python**](https://www.python.org/) is a popular open source (free) programming language and computational environment. We will use Python today to help quantify the energy and carbon footprint of generative AI LLMs. We will run Python using Spyder, an Integrated Development Environment (IDE) for Python. There are a couple of preliminary steps.

* First, open Spyder by finding and clicking on its icon on your workstation:
* Next, you will need to specify the Python interpreter we will use in this lab. In Spyder, go to **Tools > Preferences > Python Interpreter**  (Fig 2.1).



* Next, click on the folder to right of the pulldown widget, and navigate to **C/Program Files/python313/python313.exe**. Hit **Apply**.
* Close Spyder and reopen it.

Now we can explore Python a bit. To start with, we can use Python as a calculator. Type 2 + 2 and click in the text editor component of Sypder (by default, the left hand side) .

2 + 2

4

Like R –a language we will learn about later in the semester– Python is an Object Oriented Programming (OOP) language. Type:

obj1 = 2 + 2

The result, 4, can be accessed by typing the object name: obj1 (or whatever I named the object).

obj1

4

Here I create a Python numerical array of the numbers, 1, 5, and 7, with help from the Python *numpy* package:

import numpy as np  
obj2 = np.array([1,5,7])

The line import numpy as np imports *numpy*, and allows me to call the package using the name np. The code np.array calls the function array in the *numpy* package.

I can use Python objects in various ways. For instance, here I add obj1 and obj2, and take the log of the sum.

np.log(obj1 + obj2)

array([1.60943791, 2.19722458, 2.39789527])

# 3 Computation of LLM Energy and Carbon Consumption

The [EcoLogits](https://ecologits.ai/latest/tutorial/) Python library tracks the energy consumption and environmental impacts of generative AI models.

The Python code below allows evaluation of energy consumption under the [OpenAI](https://openai.com/) generative AI framework. Other frameworks, e.g., [ChatGPT](https://chat.chatbot.app/), will be comparable.

* On Lines 1 and 2, we bring in necessary components from the ecologits and openai Python modules.
* On Line 6, we specify that we want to use the OpenAI model.
* On Line 8, I supply my API[[1]](#footnote-33) key to get access to OpenAI from the command line.
* On Lines 11-14, I make a generative AI query for OpenAI and contain the results in the object response. Specifically client.chat.completions.create runs the method chat.completions.create on the object client, created on Line 7. The model argument specifies the LLM model to use. The messages argument is used to provide a list of message parameters that are required input in chat completion models. Specification of the argument is done with two pairs of character (text) strings: "role": "user" and "content": "Tell me a funny joke.". We will modify the second component of the second pair of strings in this exercise. This will change the content of the query. We will also check the energy and carbon requirements for our queries.

from ecologits import EcoLogits  
from openai import OpenAI  
import time  
  
# Initialize EcoLogits  
EcoLogits.init(providers=["openai"])  
  
client = OpenAI(api\_key="sk-proj-xIDKZtNo7UvwPdQ7-1KXcit49nqjSQ8wMUJs6H5xMIa8C0ixz0p8nolQfSJ7NWHrfR9rvEZ96\_T3BlbkFJjE0fxvSD96ojyBPhzgIaN8PUklii0P4wpojMzk6xqIRtD4r74M9Xm15rPsnf\_ghNySNZ4EAiYA")  
  
# Make a generative AI query  
response = client.chat.completions.create(  
 model="gpt-4o-mini",  
 messages=[  
 {"role": "user", "content": "Tell me a funny joke!"}  
 ]  
)

The script above is contained in the file codeecologits.py. Download this file to your computer workstation and import into Spyder using **File > Open**, or simply paste the code into the Spyder text editor.

Running the code I get:

The model architecture has not been released, expect lower precision. For further information visit https://ecologits.ai/tutorial/warnings\_and\_errors/#model-arch-not-released  
The model architecture is multimodal, expect lower precision. For further information visit https://ecologits.ai/tutorial/warnings\_and\_errors/#model-arch-multimodal

We are told that our energy consumption results may be somewhat imprecise due to a lack of specific knowledge concerning the *OpenAI* model architecture, and multimodality in the predictions.

## 3.1 Response from query

The entity response, created on Line 10-14 above, is a complex Python object with many attributes. To get the AI reply to my query: "Tell me a funny joke!", I can use:

response.choices[0].message.content

"Why don't scientists trust atoms?\n\nBecause they make up everything!"

## 3.2 Primary energy

To get “primary energy” usage (the amount of energy consumed from natural sources such as raw fuels and other forms of energy, including waste), as the result of my query, I can use:

response.impacts.pe

PE(type='PE', name='Primary Energy', value=RangeValue(min=0.0003195939443039026, max=0.0009567918742273609), unit='MJ')

Units above are MegaJoules = Joules. Recall that there are 4.184 Joules in a Calorie, and 1 Calorie is the amount of energy required to raise 1 gram of H2O 1 degree C.

## 3.3 Energy used

To get the minimum and maximum energy used in terms of direct electricity consumption of GPUs, server and other equipment from data centers, as a result of my query, I can use:

response.impacts.energy

Energy(type='energy', name='Energy', value=RangeValue(min=3.0559204e-05, max=9.355023600000002e-05), unit='kWh')

Note that energy units here are kilowatt hours.

## 3.4 CO2 produced

response.impacts.usage.energy.unit

'kWh'

To get the CO2 production equivalent resulting from the query we can use:

response.impacts.gwp

GWP(type='GWP', name='Global Warming Potential', value=RangeValue(min=1.9176027845204543e-05, max=5.7004250846415313e-05), unit='kgCO2eq')

Note that the units are kilograms of equivalent CO2.

# Questions

1. In Python, create a numerical object with your name and perform some simple math with it.
2. Create a simple OenAI query by modifying the fourth item in the message argument on Line 13 in the Python code above.
   1. Show the AI reply by running: response.choices[0].message.content.
   2. Run response.impacts.energy. How many MegaJoules of primary energy were used?
3. According to WebMD, a moderately active female between the ages of 19-24 should consume 2,000-2,200 kcal (kilocalories) per day.
   1. Compute, using Python or some other software, the number of MegaJoules in 2000 kcal. Mathematically, you could use an approach like:
   * Show your work using screen snips.
   1. Your computation in Question 1 probably took less than a second. To get the approximate amount of energy *you* consumed in 1 second, divide your result in (2a) by . That is, hrs minutes seconds.
   2. Compare your answers in (1b) and (2b). What are your conclusions?
4. We now consider the moderately difficult (AI-generated) query: “Explain the most likely reason why a friend would unexpectedly bring a freshly baked loaf of bread to a dinner party. Your explanation should take into account the social context, the act of baking, and the expectations of a host. Furthermore, provide a potential, but less likely, alternative reason and contrast the two.”
   1. Insert the code x = time.time() to a line *above* where the response object is created in the main OpenAI Python script. For instance, you could insert the code on Line 9. Also, add the script y = time.time() to a *new line*, below the current last line of the script (below Line 16). Show a screenshot of your work.
   2. Run the entire Python script (including the newly inserted code) and get the system time required for the query by typing: y - x.
   3. Get the energy used by the query by typing: response.impacts.energy.
   4. To consider the energy required for a server to handle a constant daily stream of moderate (silly) threads like these, calculate the kilowatt hours required if the query thread above took an entire day. To do this, let be the number of seconds required by your query thread (your answer in 3b), and let be the energy used (your answer in 3c). Calculate:
   * Note that a single commercial server can have hundreds of cores and be capable of handling thousands of threads in parallel. Server farms can contain millions of servers, with the potential for processing billions of threads simultaneously.
   1. Comment on your result in (3d) given that an average US household uses 37.5 kWh/day, and households in East Africa use about 3.7 kWh/day.

# References

Bushwick, Sophie. 2023. “What the New GPT-4 AI Can Do.” *Scientific American* 16.

De Vries, Alex. 2023. “The Growing Energy Footprint of Artificial Intelligence.” *Joule* 7 (10): 2191–94.

Kolbert, Elizabeth. 2024. “The Obscene Energy Demands of AI.” *The New Yorker* 9.

Russell, Stuart J, and Peter Norvig. 2021. “Artificial Intelligence: A Modern Approach, Global Edition 4e.”

Van Der Vlist, Fernando, Anne Helmond, and Fabian Ferrari. 2024. “Big AI: Cloud Infrastructure Dependence and the Industrialisation of Artificial Intelligence.” *Big Data & Society* 11 (1): 20539517241232630.

Wikipedia. 2025a. “Artificial Intelligence.” <https://en.wikipedia.org/wiki/Artificial_intelligence>.

———. 2025b. “Hallucination (Artificial Intelligence).” <https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence)>.

———. 2025c. “Large Language Model.” <https://en.wikipedia.org/wiki/Large_language_model>.

———. 2025d. “Transformer (Deep Learning Architecture).” <https://en.wikipedia.org/wiki/Transformer_(deep_learning_architecture)>.

Zack, Travis, Eric Lehman, Mirac Suzgun, Jorge A Rodriguez, Leo Anthony Celi, Judy Gichoya, Dan Jurafsky, et al. 2024. “Assessing the Potential of GPT-4 to Perpetuate Racial and Gender Biases in Health Care: A Model Evaluation Study.” *The Lancet Digital Health* 6 (1): e12–22.

1. API stands for Application Programming Interface. An API provides glue code that allows one software framework (e.g., Python) to work directly with a foreign software system (e.g., OpenAI). [↑](#footnote-ref-33)