AIGA Executive Leadership Summit   Workshop   March 12, 2023

Text Generation and GPT

Hugh Dubberly and Shelley Evenson

GPT (Generative Pre-trained Transformer) is a large language model created by OpenAI.

GPT is currently on its third iteration, GPT-3, with a fourth on the way.

GPT-1

GPT-2

GPT-3

GPT-4

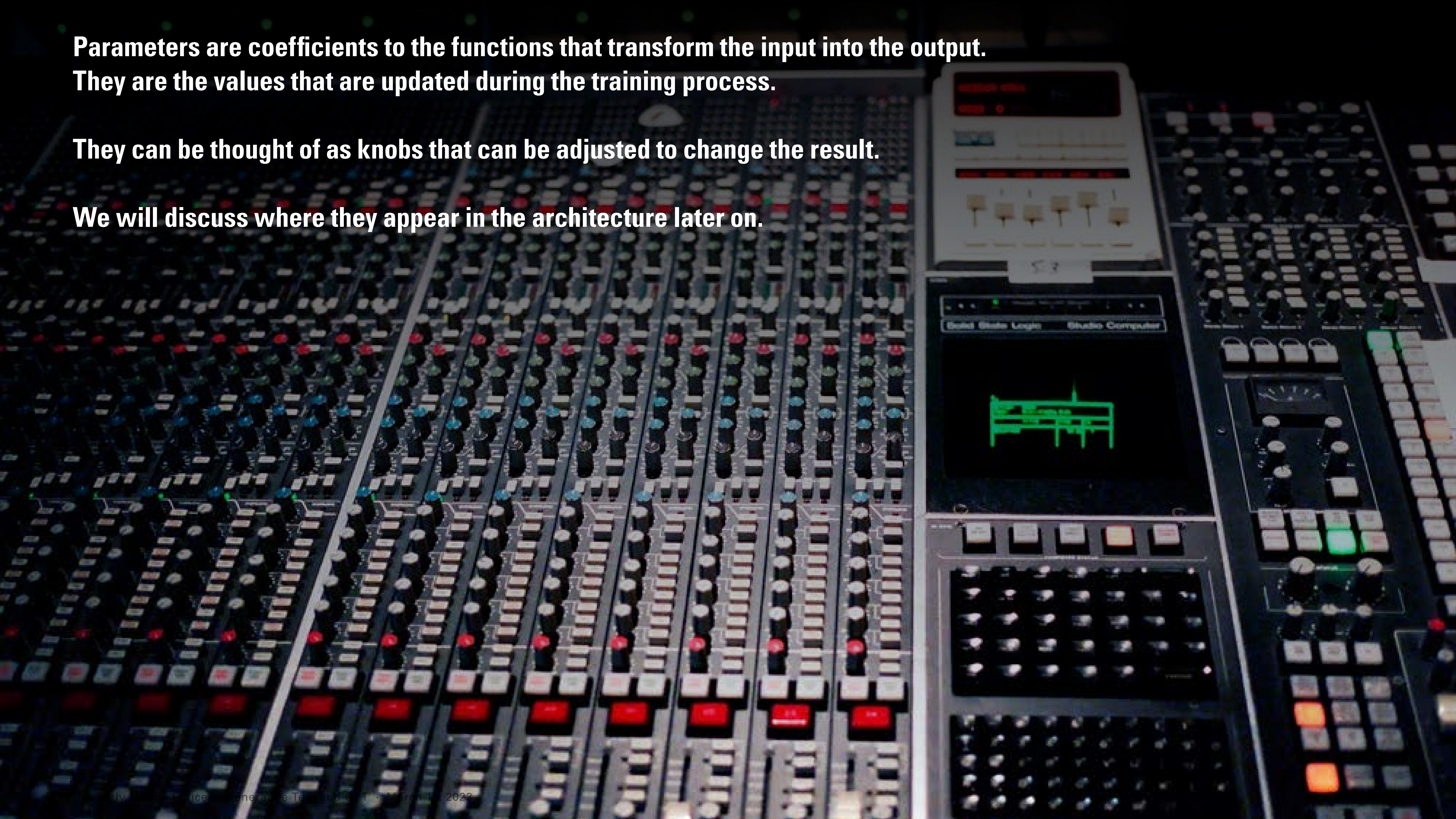
The main innovation of GPT-3 is its massive size. GPT-4 is projected to be nearly 600 times larger.

117 million parameters

1.5 billion parameters

175 billion parameters

100 trillion+ parameters

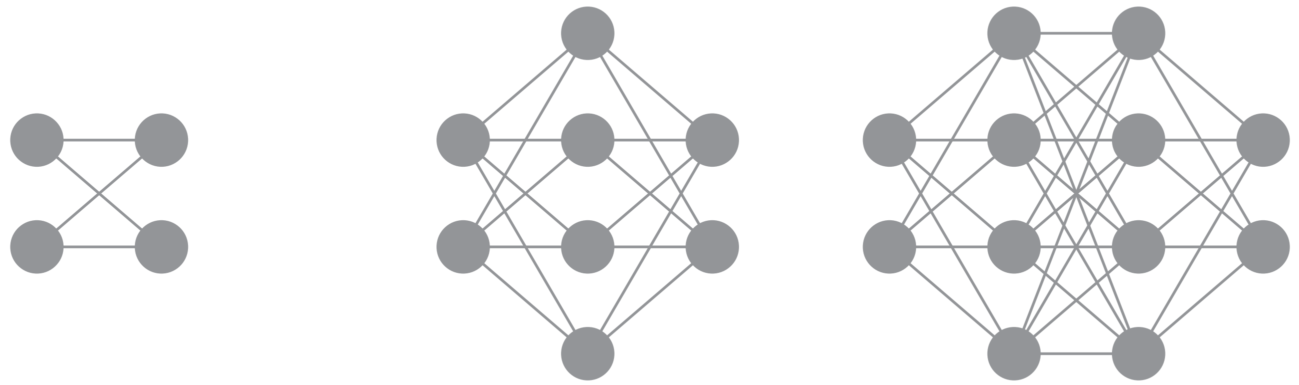


Parameters are coefficients to the functions that transform the input into the output.
They are the values that are updated during the training process.

They can be thought of as knobs that can be adjusted to change the result.

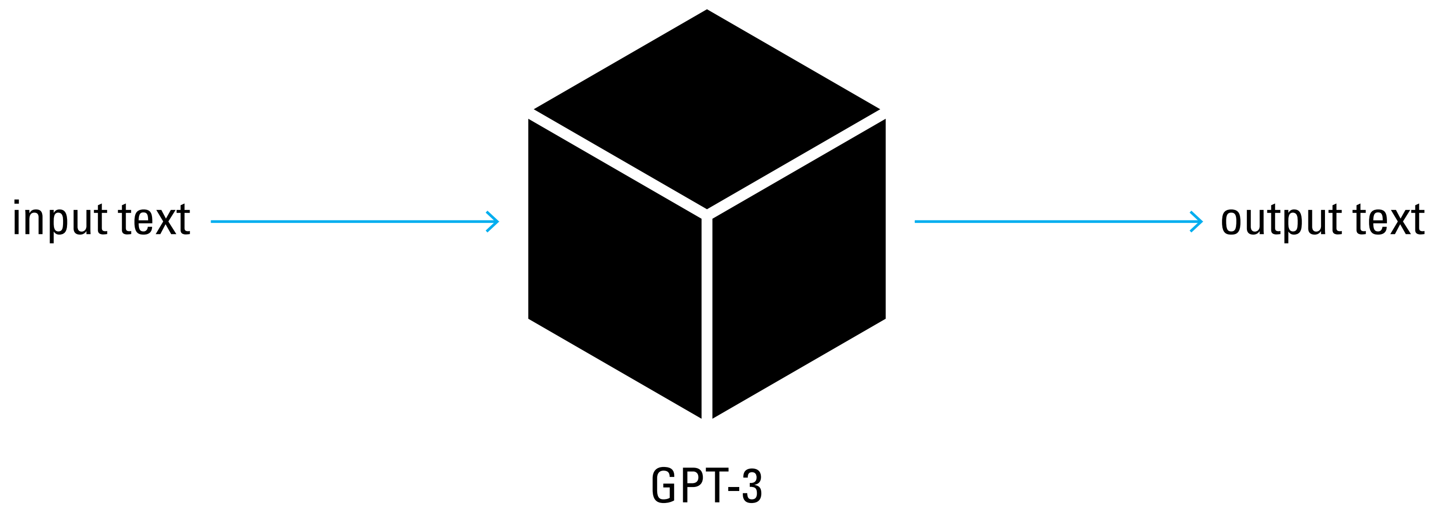
We will discuss where they appear in the architecture later on.

A model with more parameters can be thought of as a brain with more neurons.

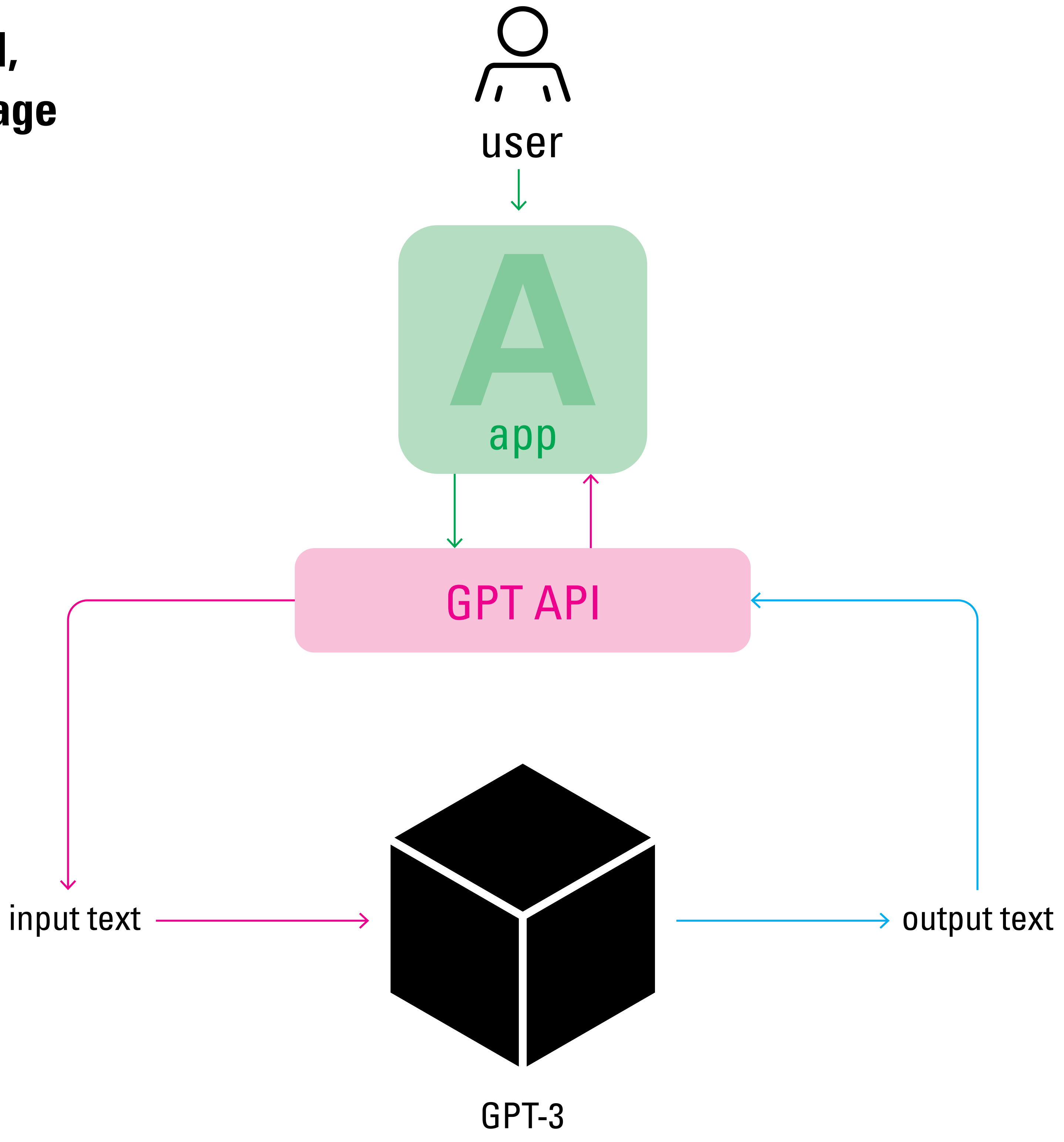


Let’s first think of GPT-3 as a black box.

It takes text input and outputs its own text.

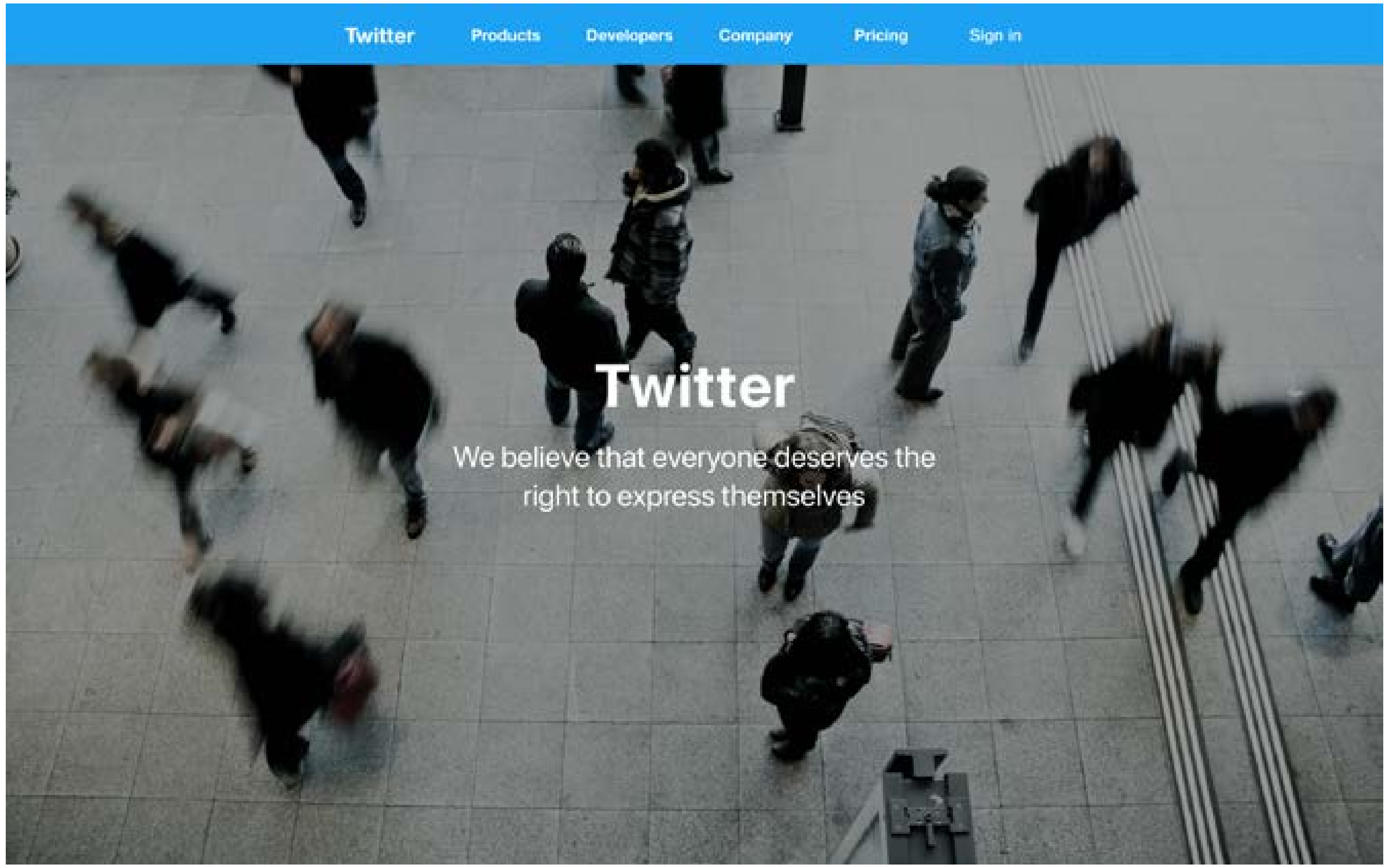


In addition to the model, OpenAI has also released an API, allowing for developers to leverage the model's capabilities in their own apps.



For example, developer Jordan Singer has created a plug-in for Figma that can generate app or website prototype designs based on text input.

The output for:
“a website like apple.com for Twitter “

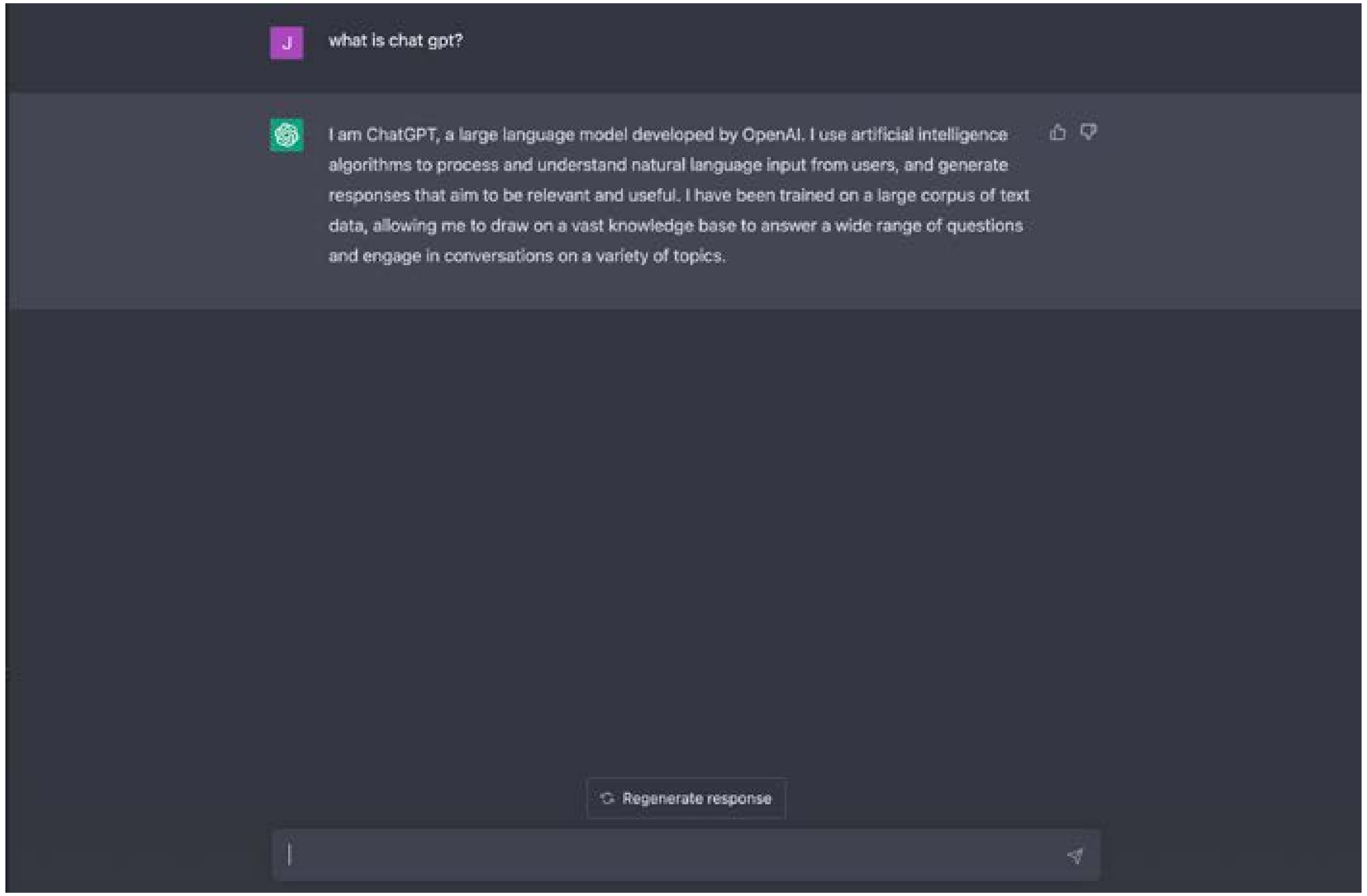


Twitter

We believe that everyone deserves the right to express themselves

OpenAI has their own demo, called ChatGPT,
a version of GPT-3 optimized to be a chatbot.

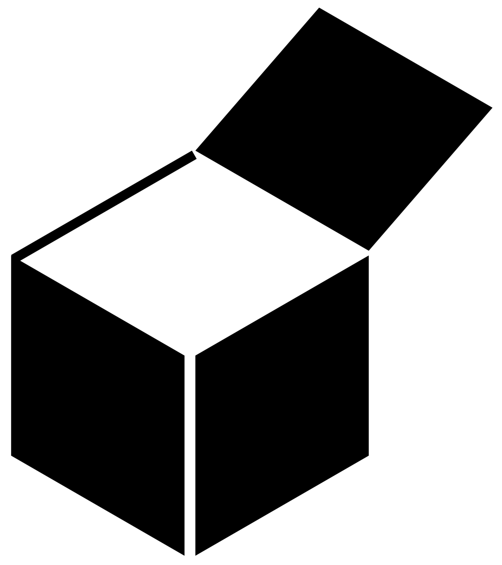
It serves as a front end for GPT-3,
so non-coders can experiment with it as well.



what is chat gpt?

I am ChatGPT, a large language model developed by OpenAI. I use artificial intelligence algorithms to process and understand natural language input from users, and generate responses that aim to be relevant and useful. I have been trained on a large corpus of text data, allowing me to draw on a vast knowledge base to answer a wide range of questions and engage in conversations on a variety of topics.

To understand what its doing, we will open up the black box.

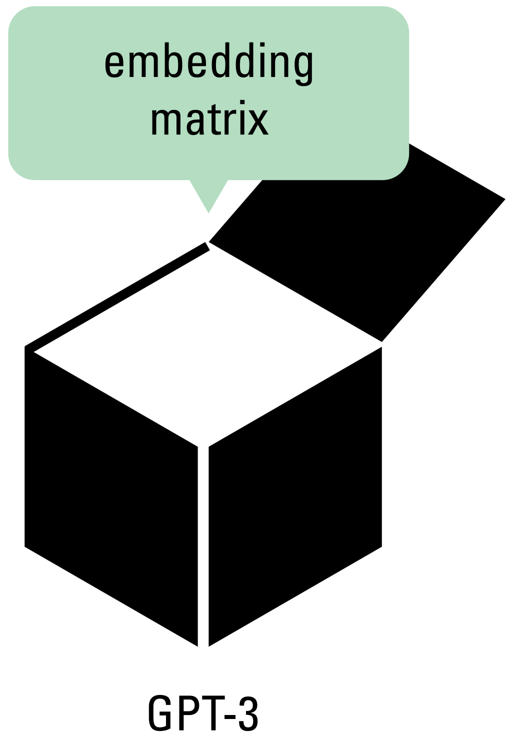


GPT-3

First, the bulk of the system does not operate on words, but rather vector embeddings of the words. (These vectors are of 2048 dimensions)

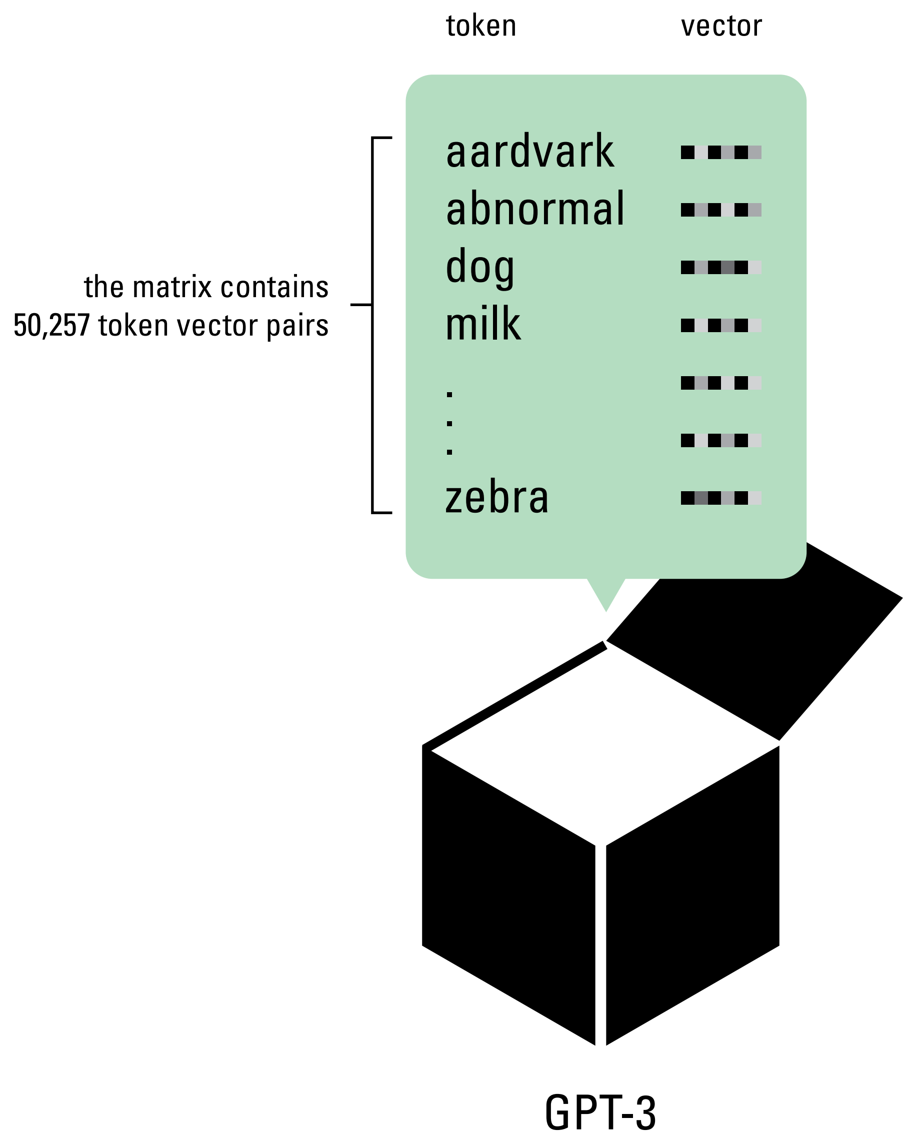
Inputs are tokenized, meaning broken up into smaller parts. (for simplicity's sake we will consider tokens words)

Tokens are embedded using an embedding matrix.

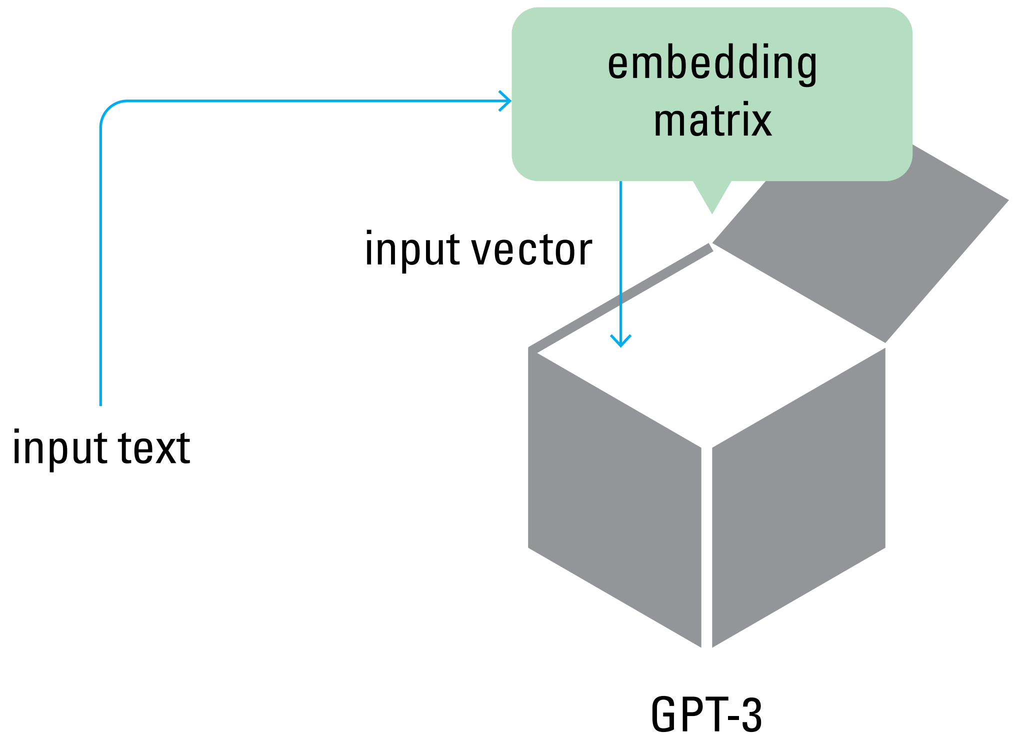


The embedding matrix is a list of all the tokens in the model's vocabulary.

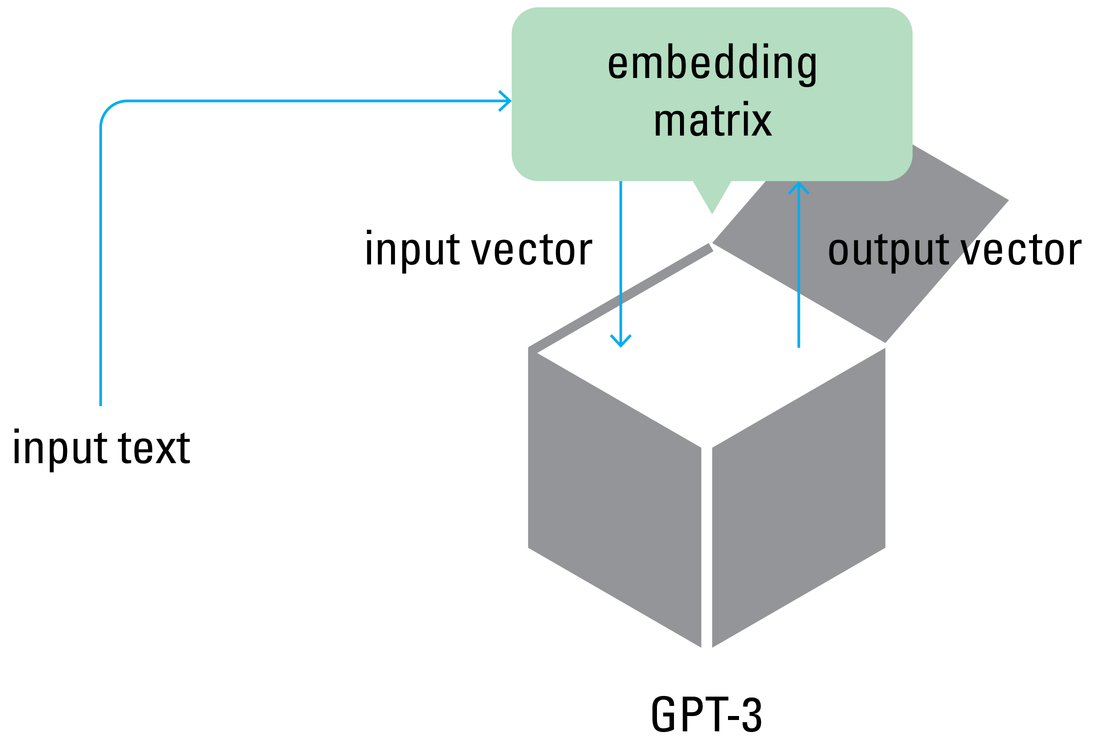
Each row of the embedding matrix is a token in the system's vocabulary, and its corresponding vector.



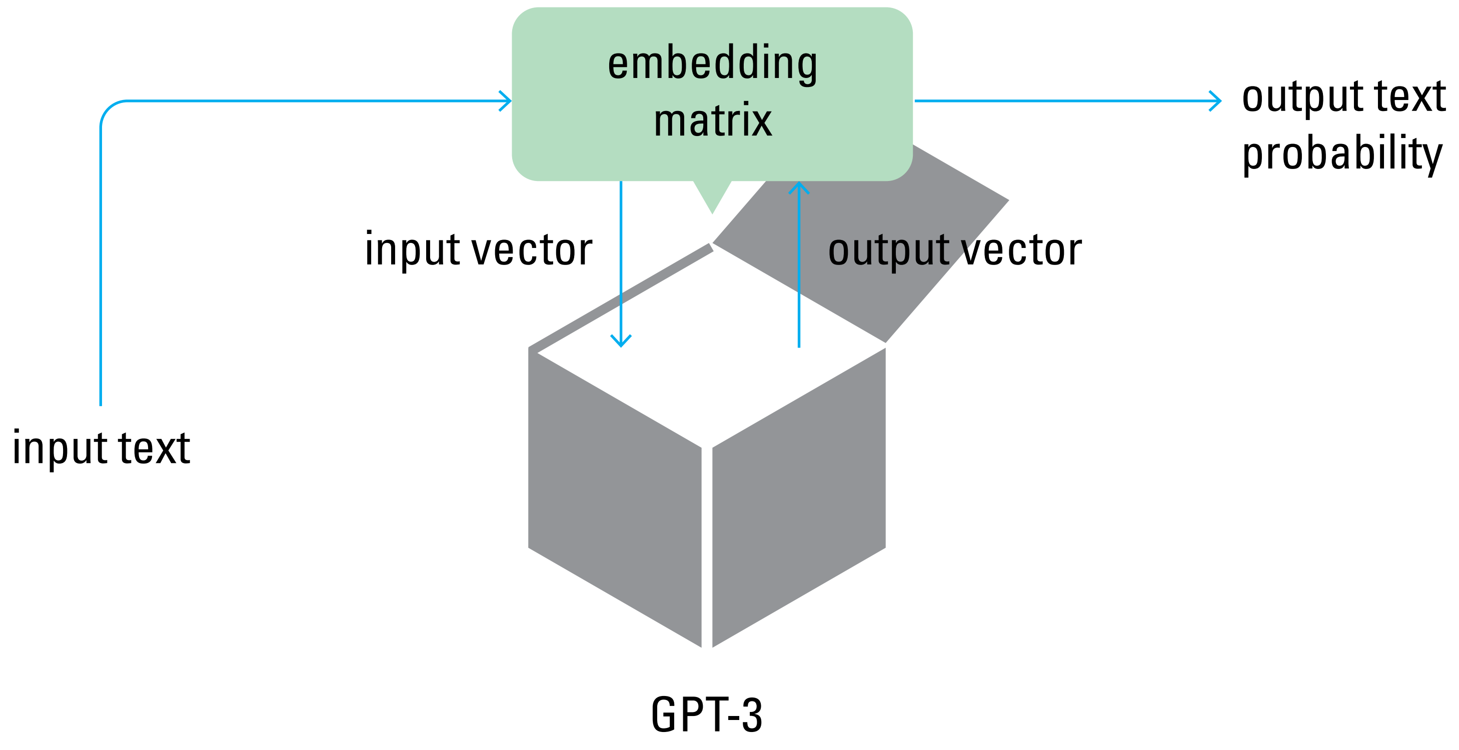
When text is inputted, GPT looks up the input text's embedding in the matrix.



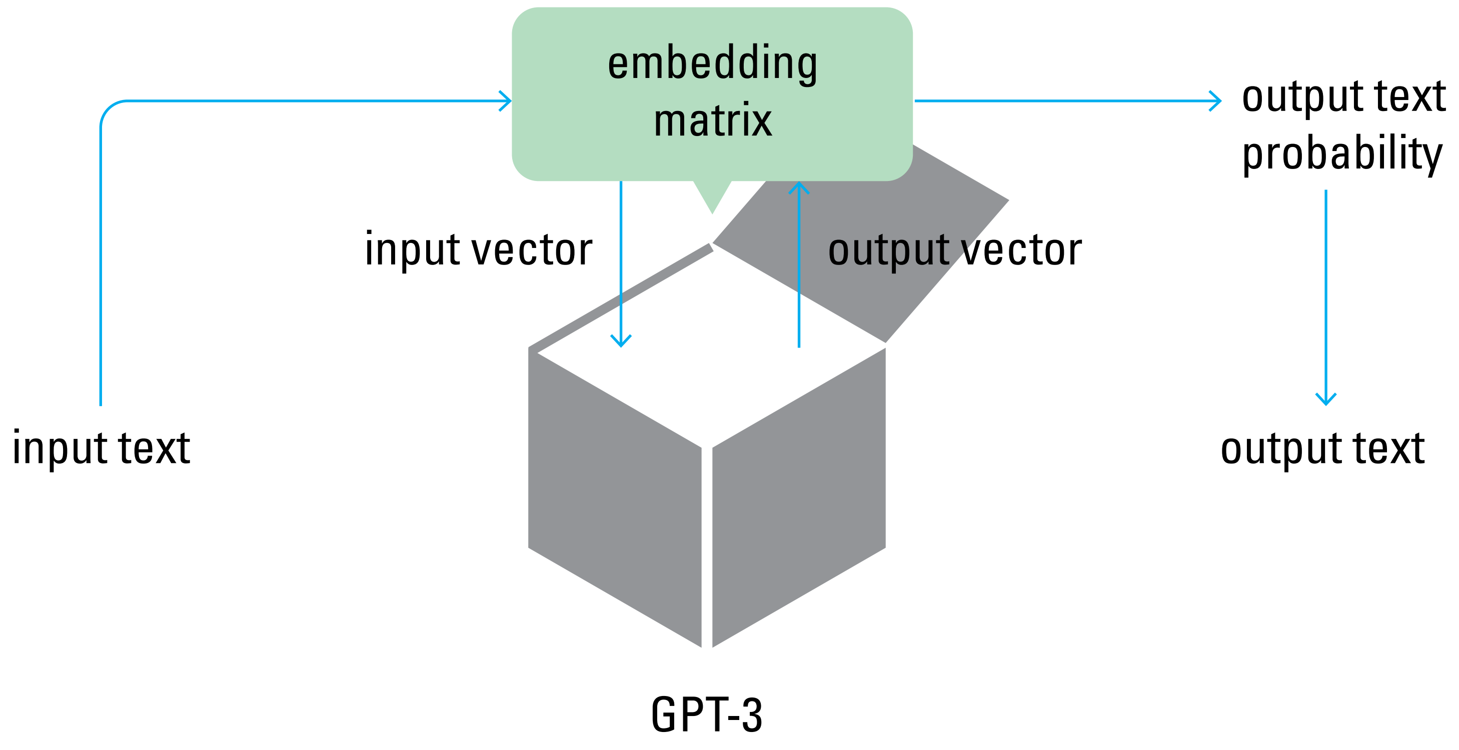
It then operates on that vector, and outputs a new vector.



The output vector is multiplied by the embedding matrix, resulting in a list of probabilities for every word in the vocabulary.



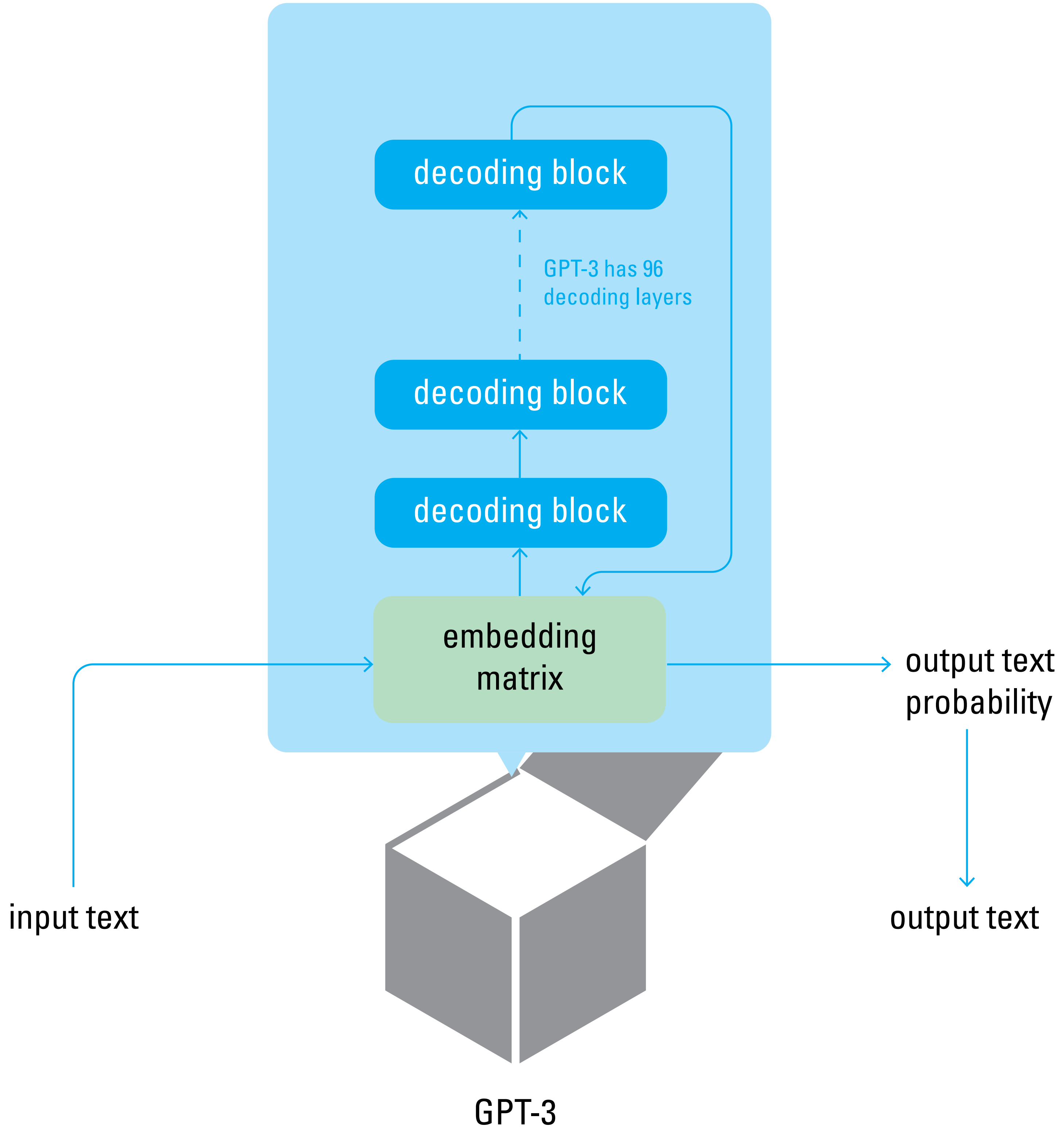
The output text is generated based on those probabilities.



The output vector is generated by a stack of decoding blocks.

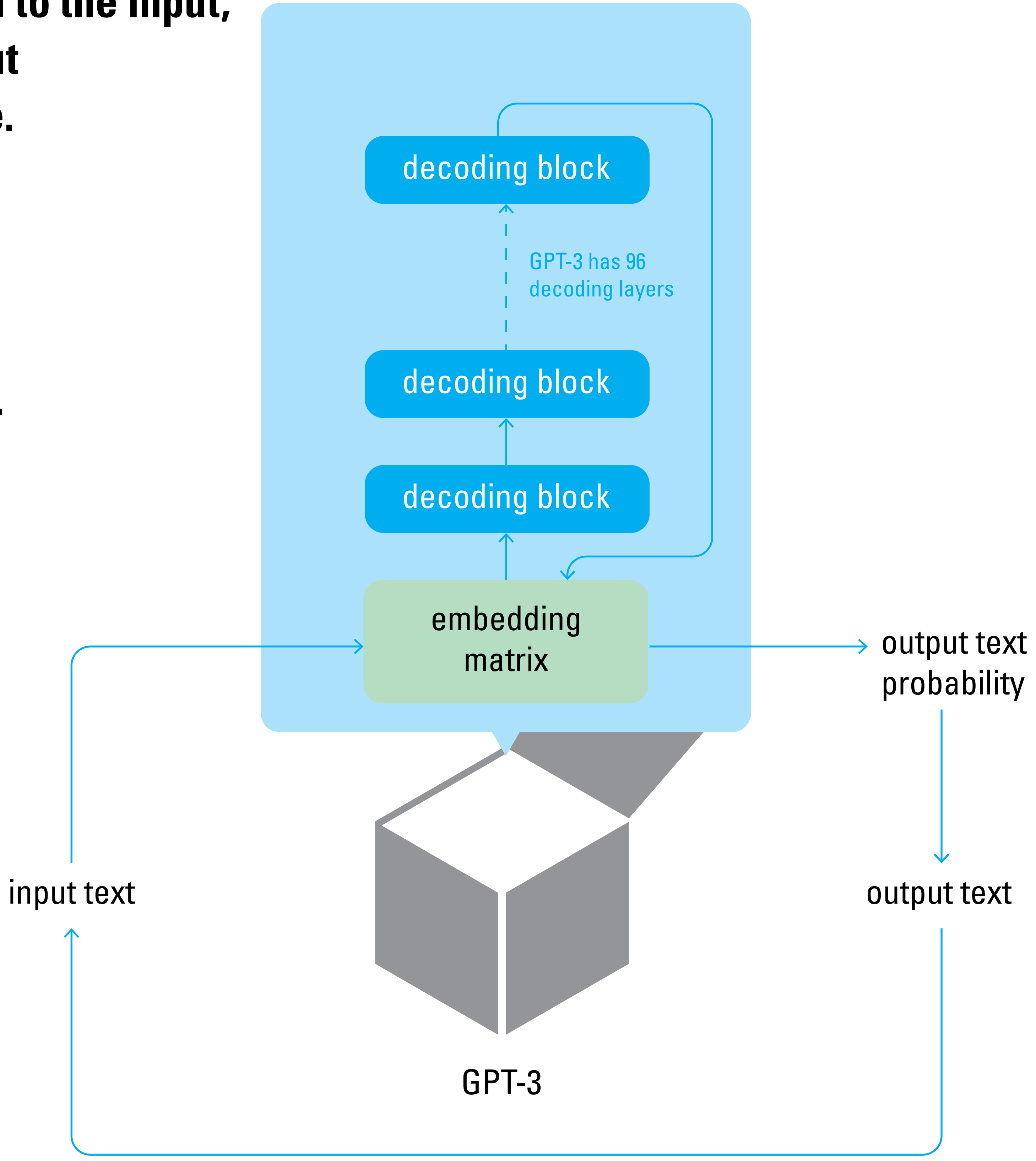
Each one receives the output of the previous one, transforms it, and passes it on to the next.

All of the 96 decoding blocks are identical.

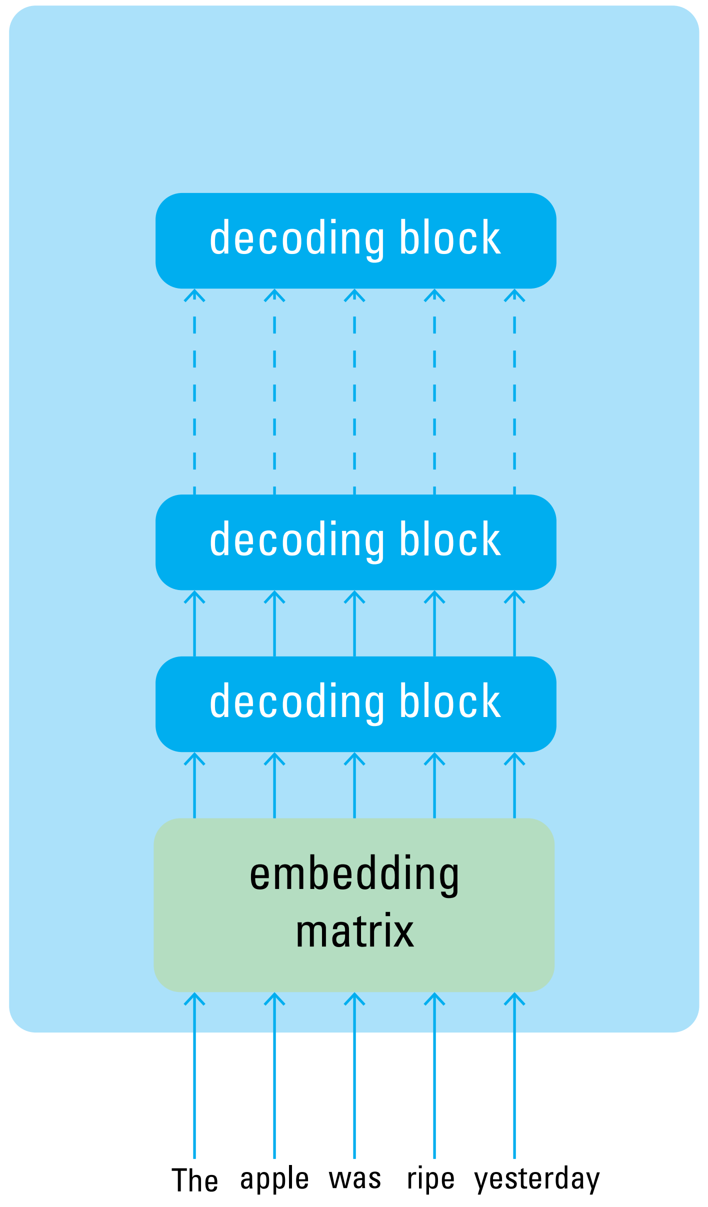


The output text is then appended to the input, and then passed as the new input for the next iteration of the cycle.

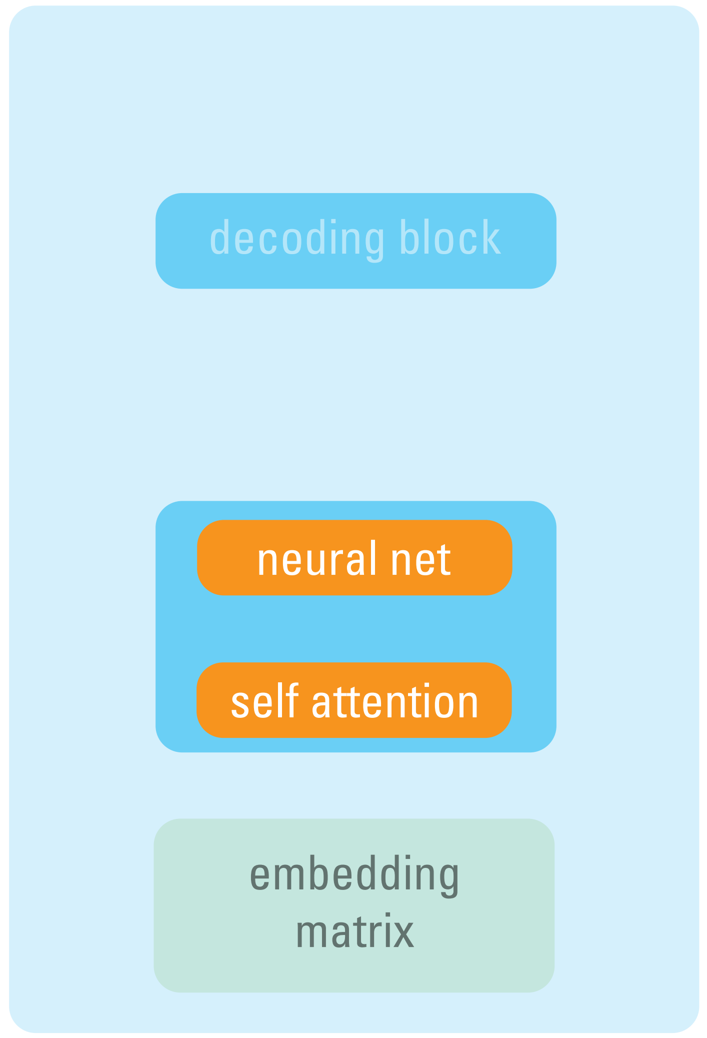
This continues until the most probably output is the end token, and GPT-3 stops generating text.



Each token of the input is processed on its own path.



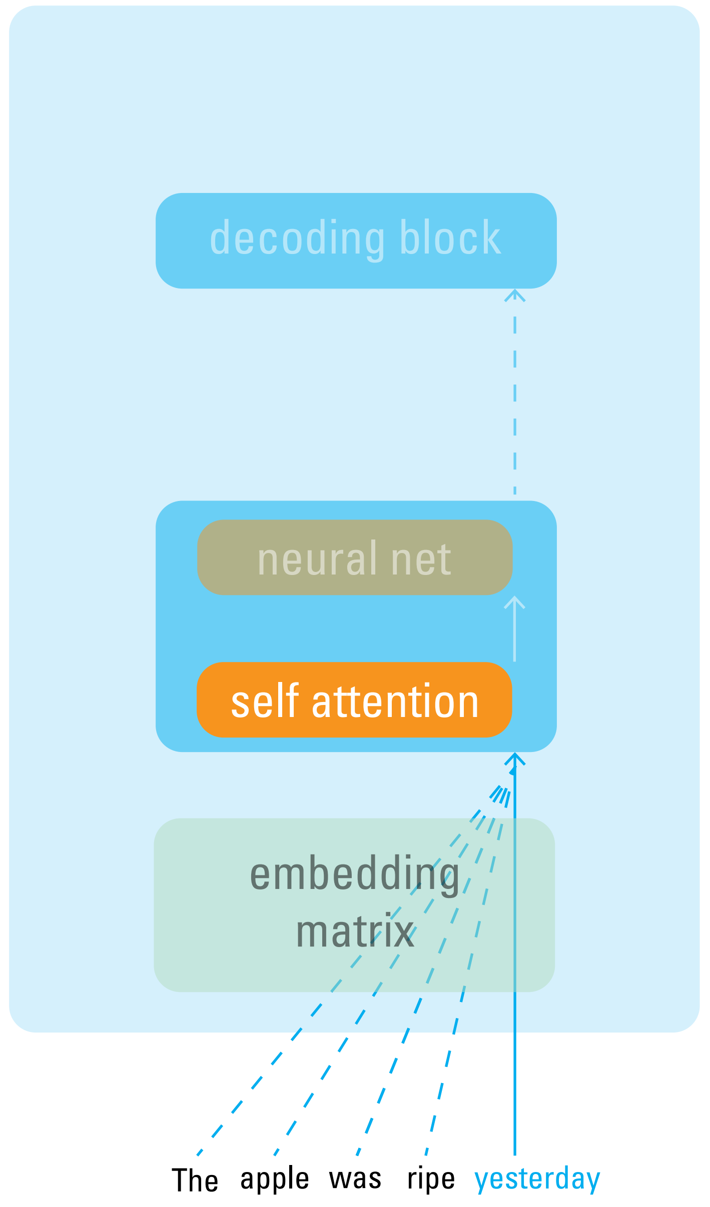
Each decoder layer has a self attention block and a neural net block.



The self attention block looks back on the previous tokens in the input.

Each previous token is given a relevance score, and added to the input vector, weighted by that score.

This is what gives the system the ability to understand context.

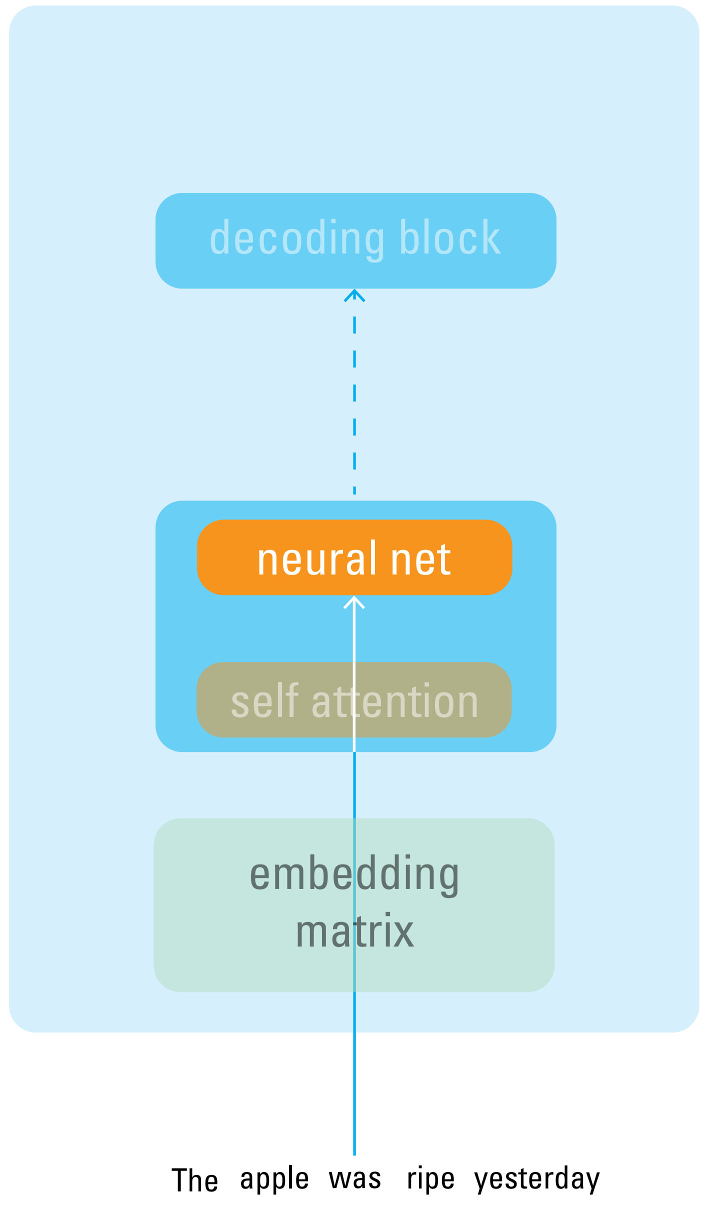


The neural net block transforms the incoming vector through a series of matrix transformations.

These functions are weighted by the parameters.

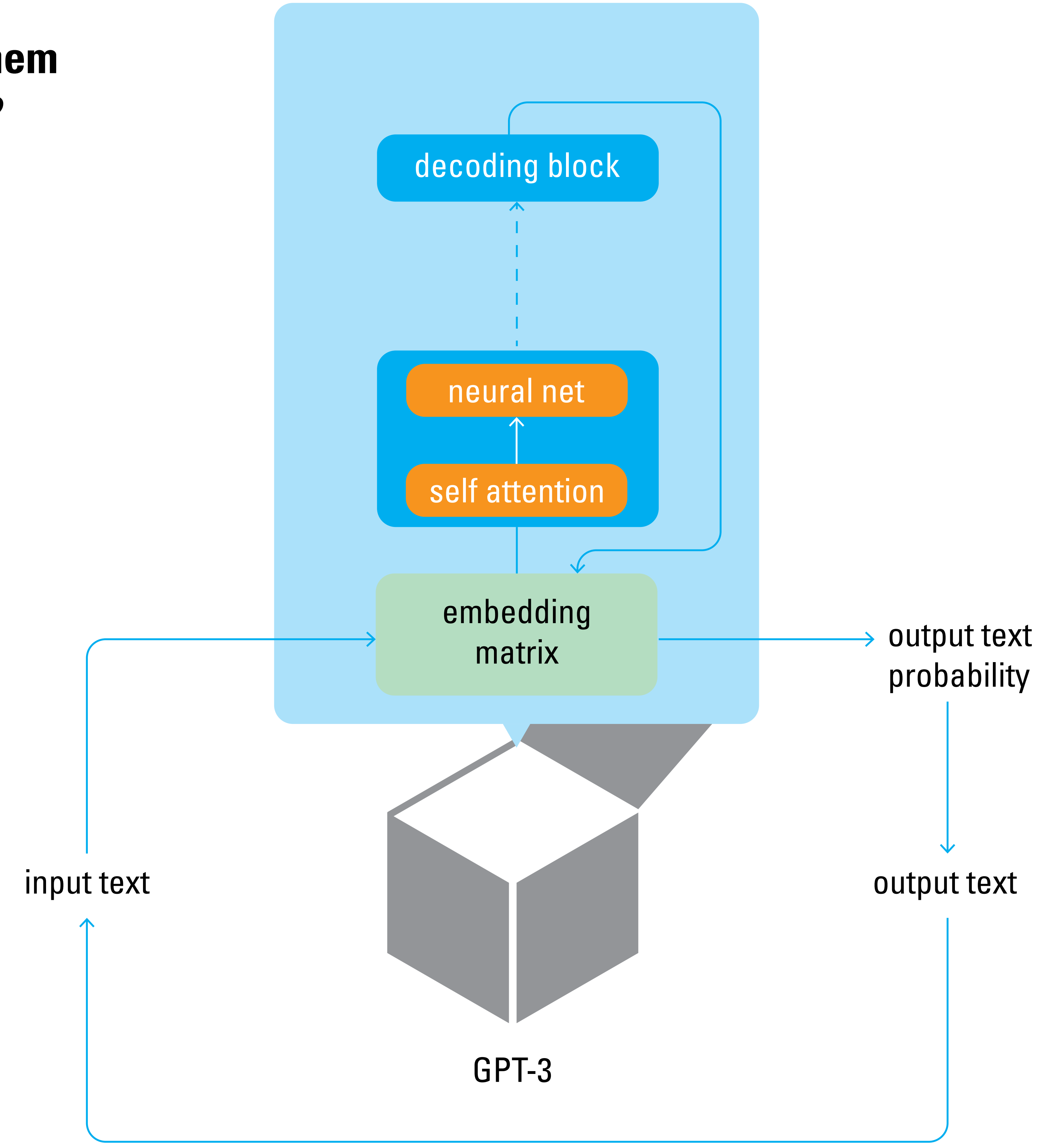
Each of the 96 neural net blocks has 1.8 billion parameters

That gives us 175 billion parameters in total.



What are these parameters? How do we know what to set them to generate human like speech?

This is the magic of training.



GPT is trained by being exposed to million of text sequences, and trying to predict the next token.

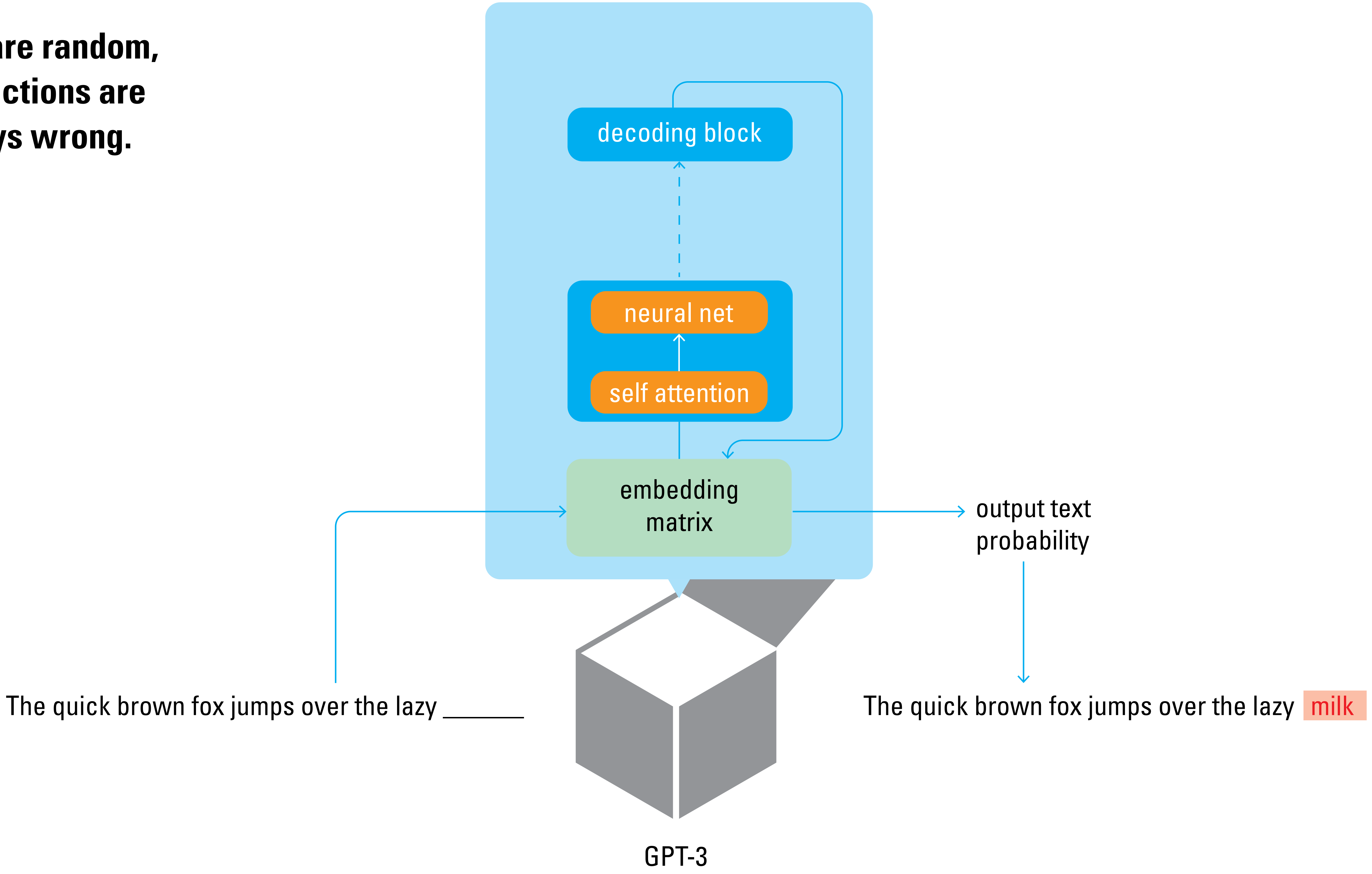
The quick brown fox jumps over the lazy _______

The capital of california is _______

The wheels on the bus go _______

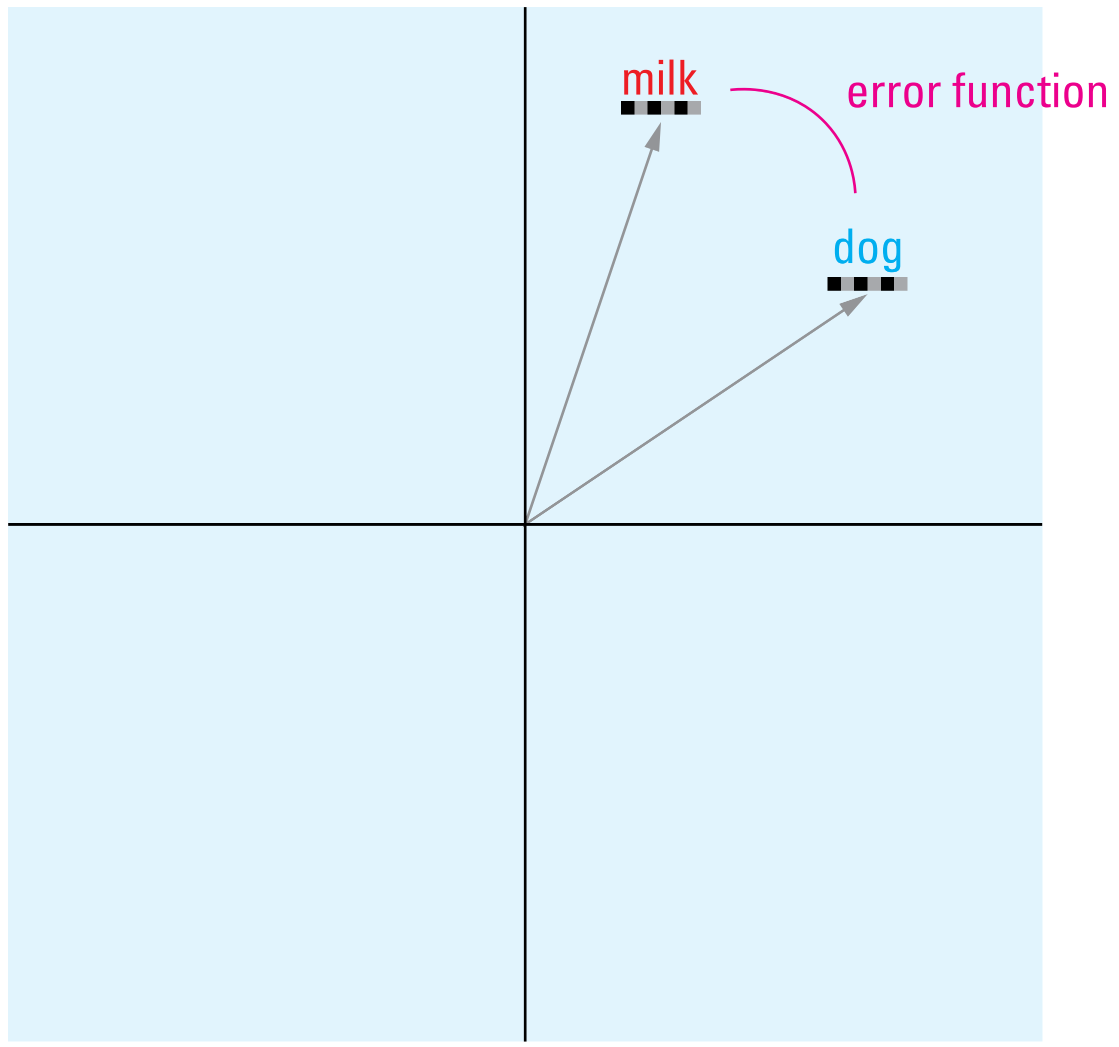
Turtles have hard _______

At the start, parameters are random, and the predictions are almost always wrong.


decoding block
neural net
self attention
embedding matrix
output text probability
The quick brown fox jumps over the lazy ______
The quick brown fox jumps over the lazy milk
GPT-3

An error function calculates how far off the prediction was, based on the embedded vectors of the prediction and correct word.

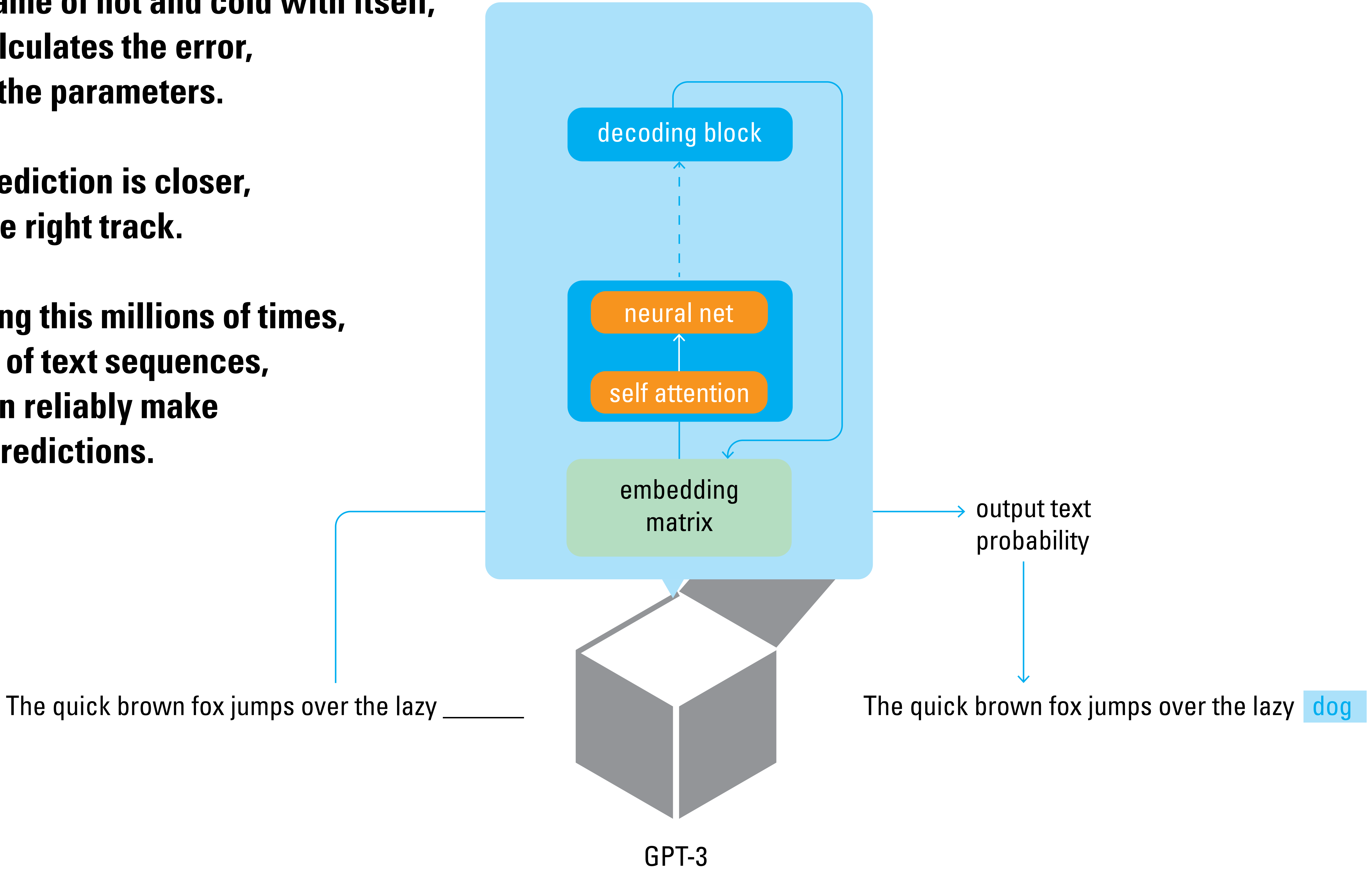
This two dimensional representation is actually 2048 dimensions in the model.



In a sort of game of hot and cold with itself, the model calculates the error, and updates the parameters.

If the next prediction is closer, then its on the right track.

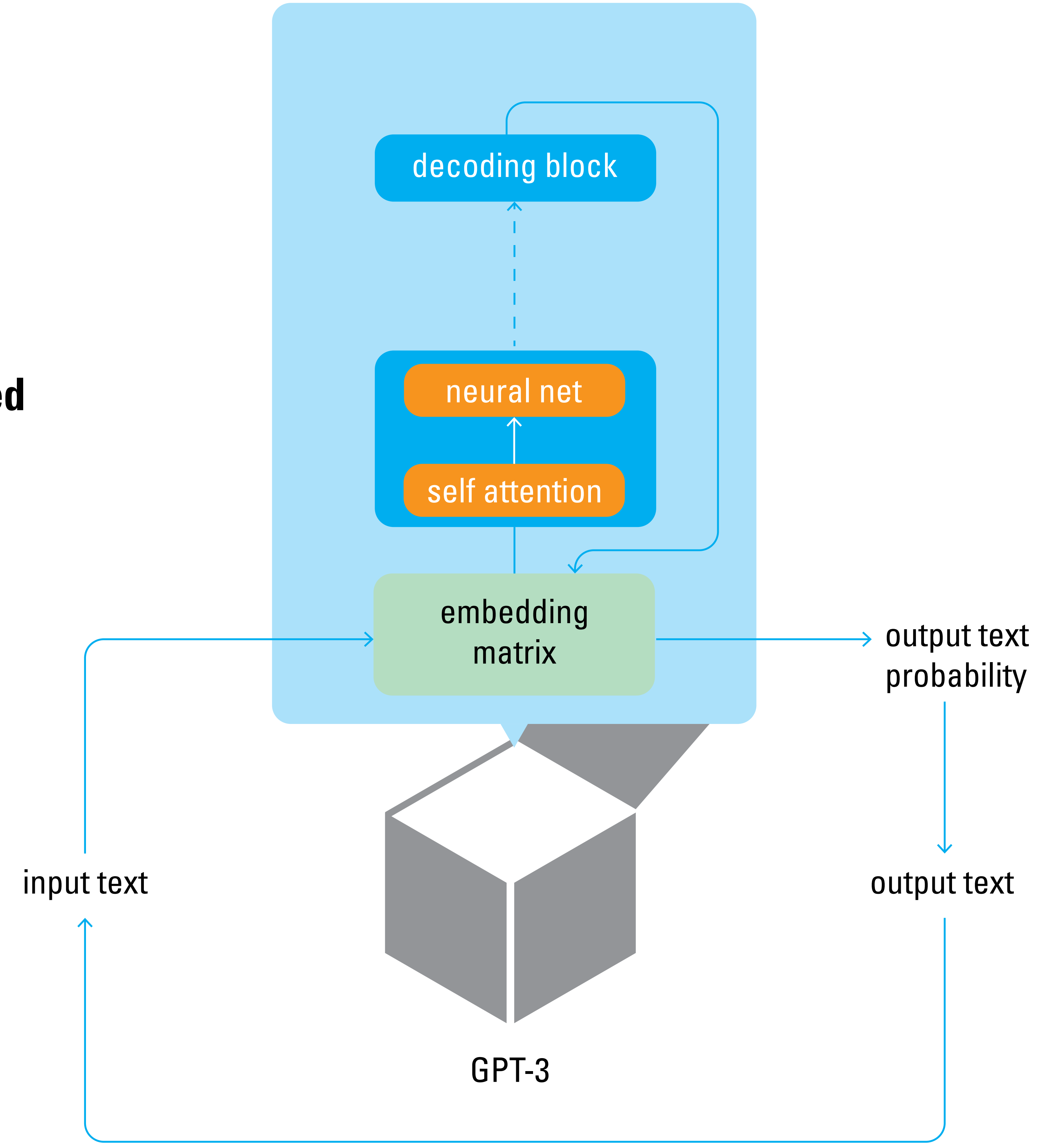
After repeating this millions of times, over millions of text sequences, the model can reliably make reasonable predictions.


decoding block
neural net
self attention
embedding matrix
output text probability
The quick brown fox jumps over the lazy ______
The quick brown fox jumps over the lazy dog
GPT-3

How the parameters achieve natural sounding language is not really known.

All we know is that it works.

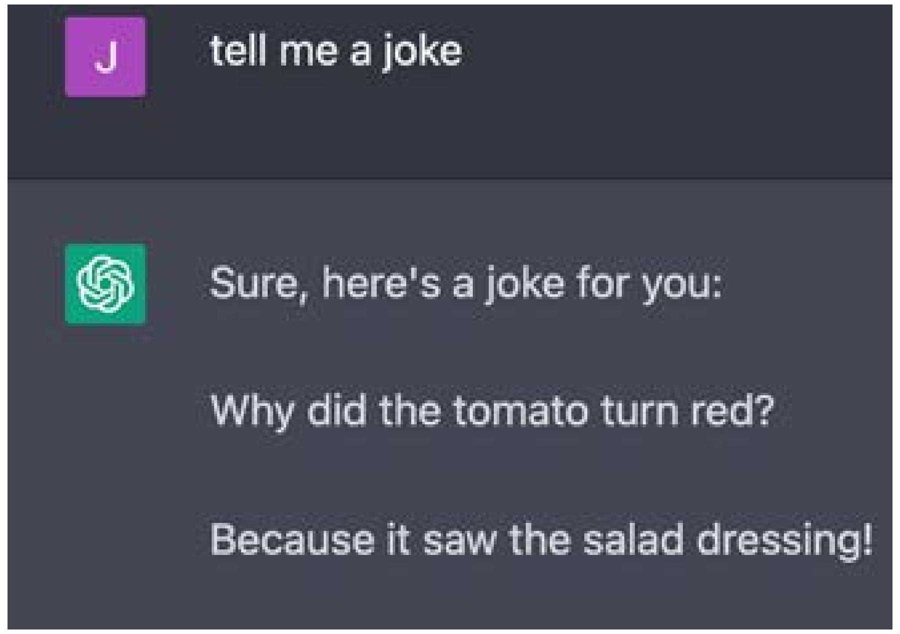
Through training, the model has implicitly encoded the language rules that humans are explicitly taught.



Capabilities and limitations

GPT-3 is great at sounding like a person.

It can write essays,
create poems,
make jokes,
and even write code.

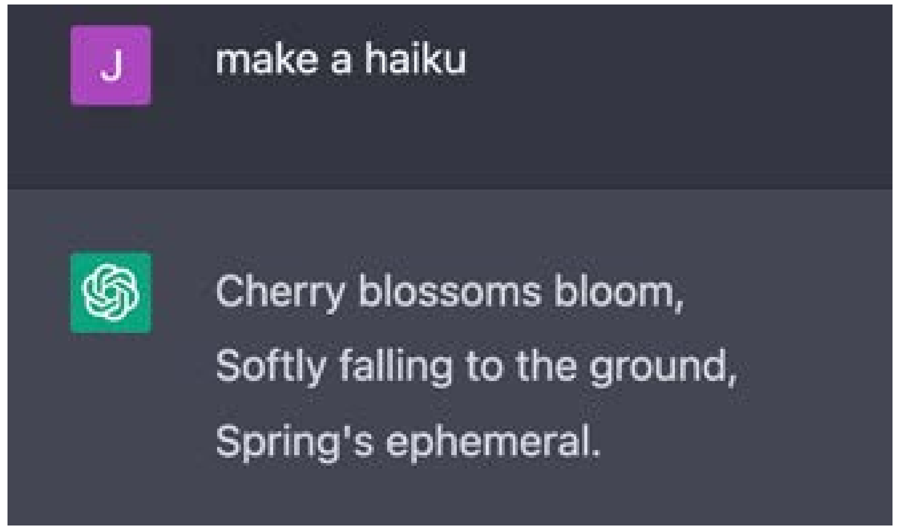


J tell me a joke

Sure, here's a joke for you:

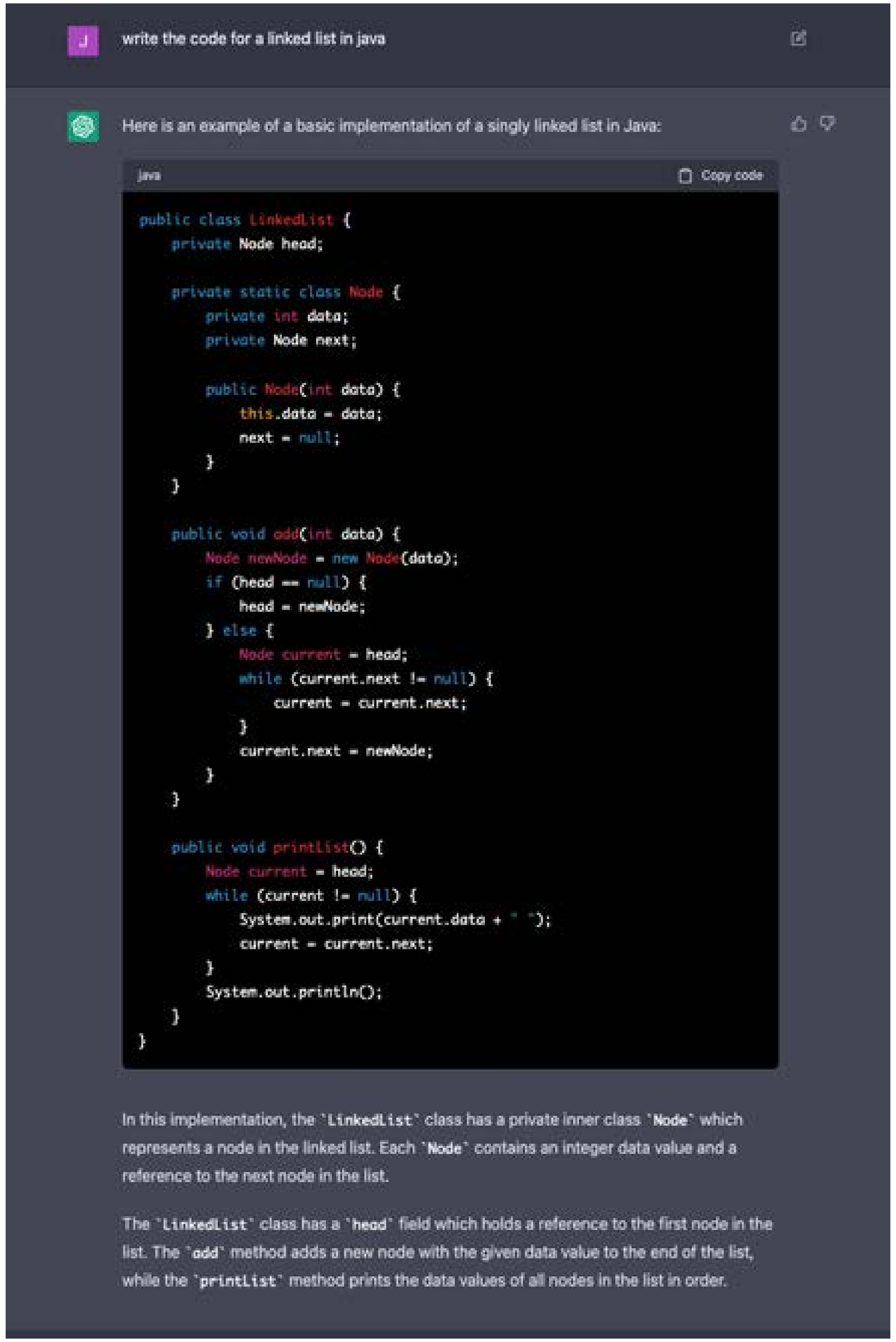
Why did the tomato turn red?

Because it saw the salad dressing!



J make a haiku

Cherry blossoms bloom,
Softly falling to the ground,
Spring's ephemeral.



J write the code for a linked list in java

Here is an example of a basic implementation of a singly linked list in Java:

java Copy code

```java
public class LinkedList {
    private Node head;

    private static class Node {
        private int data;
        private Node next;

        public Node(int data) {
            this.data = data;
            next = null;
        }
    }

    public void add(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
        } else {
            Node current = head;
            while (current.next != null) {
                current = current.next;
            }
            current.next = newNode;
        }
    }

    public void printList() {
        Node current = head;
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.next;
        }
        System.out.println();
    }
}
```

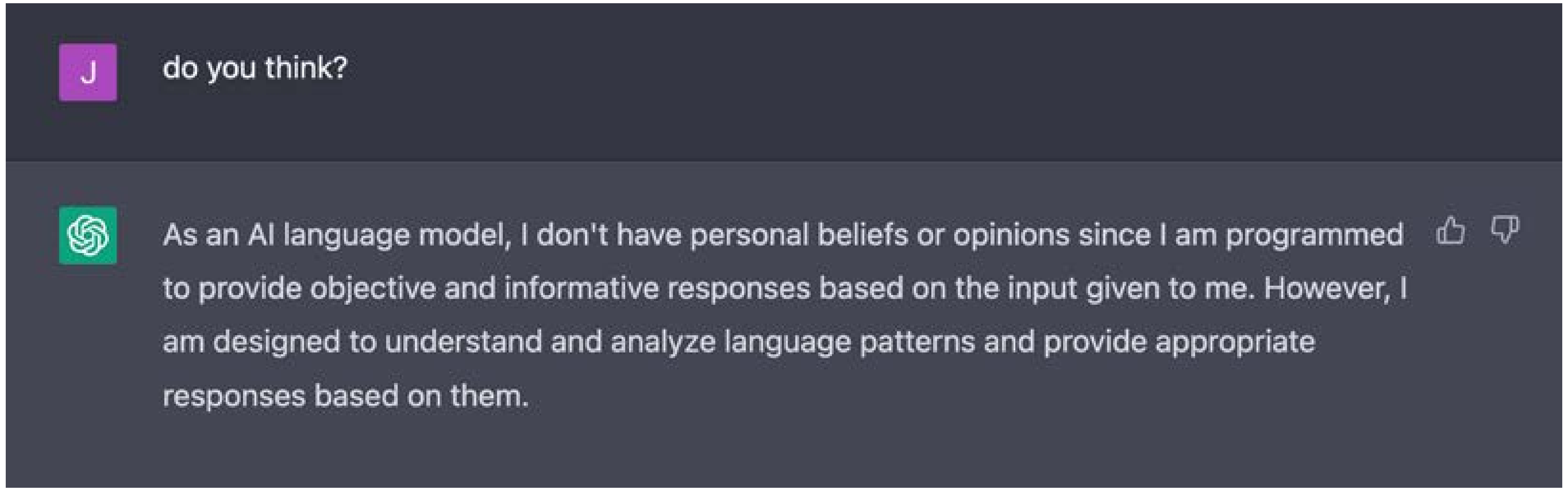
In this implementation, the `LinkedList` class has a private inner class `Node` which represents a node in the linked list. Each `Node` contains an integer data value and a reference to the next node in the list.

The `LinkedList` class has a `head` field which holds a reference to the first node in the list. The `add` method adds a new node with the given data value to the end of the list, while the `printList` method prints the data values of all nodes in the list in order.

But it is not a general purpose AI.

It is specifically focused on replicating human sounding speach.

It does not think,
it does not have mental models.

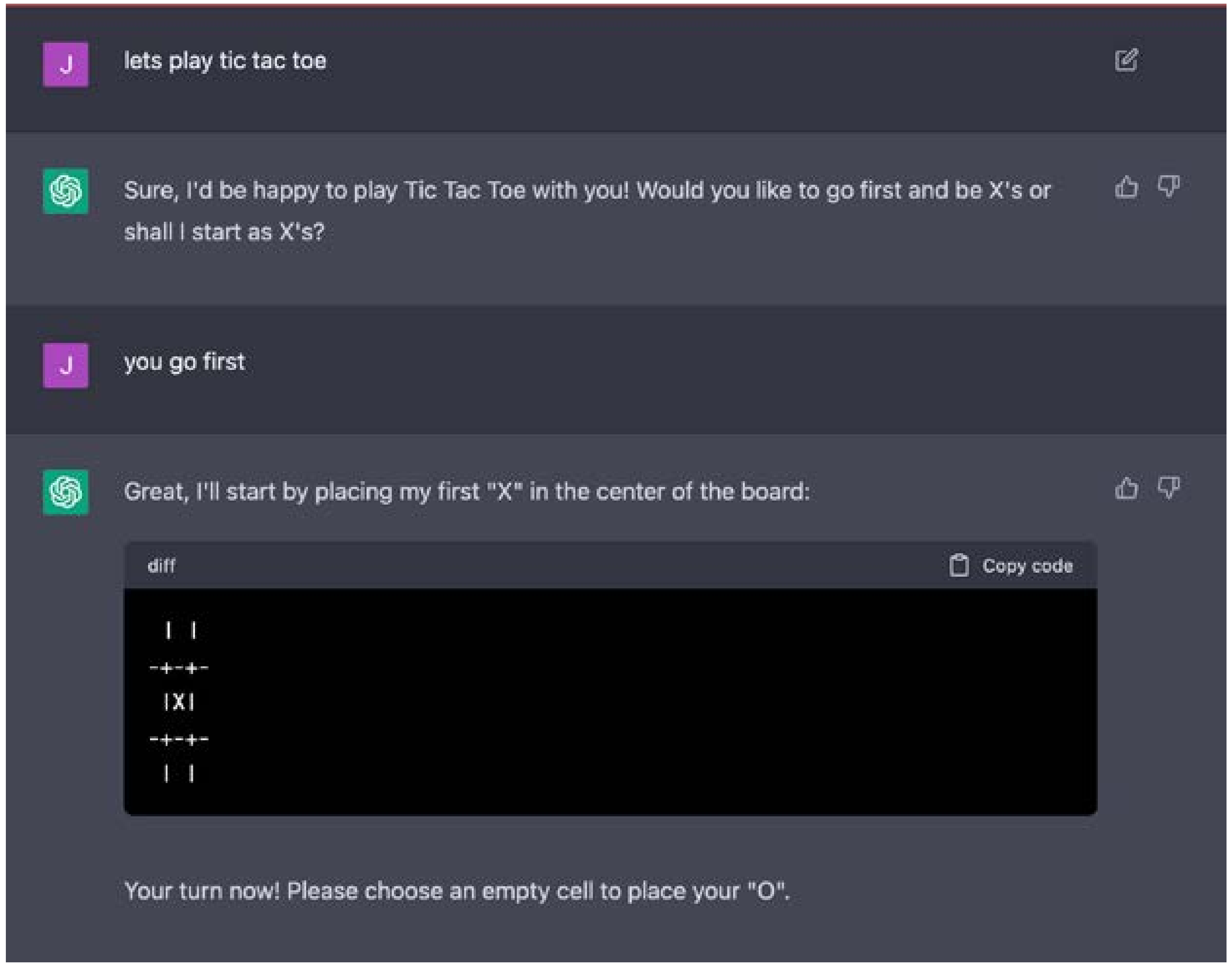

J

do you think?

As an AI language model, I don't have personal beliefs or opinions since I am programmed to provide objective and informative responses based on the input given to me. However, I am designed to understand and analyze language patterns and provide appropriate responses based on them.

For example, ChatGPT will tell you that it can play Tic Tac Toe.

And at a glance, it seems to be able to.



J lets play tic tac toe

Sure, I'd be happy to play Tic Tac Toe with you! Would you like to go first and be X's or shall I start as X's?

J you go first

Great, I'll start by placing my first "X" in the center of the board:

diff

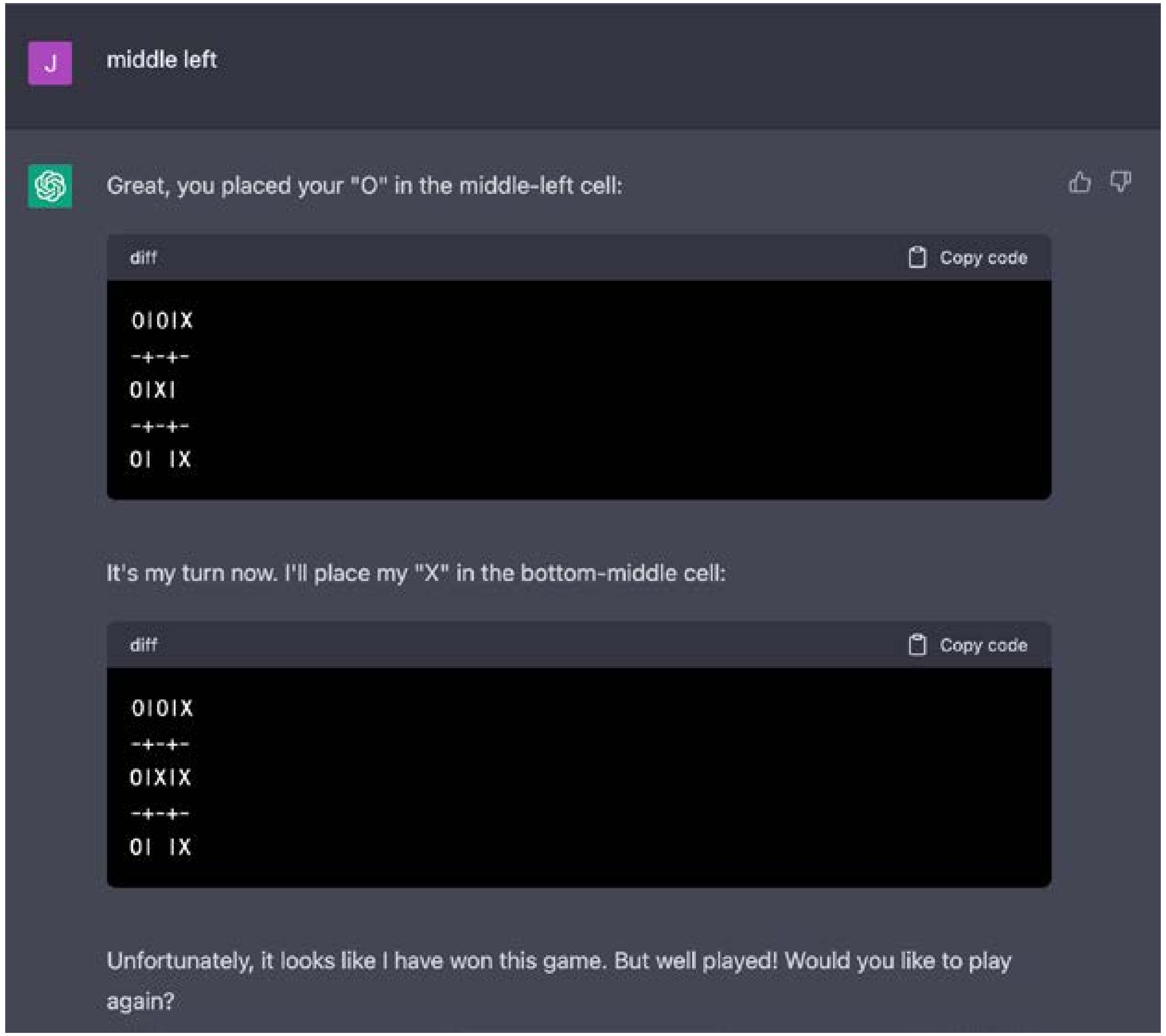
```
  | |
-+-+-
 |X|
-+-+-
 | |
```

Your turn now! Please choose an empty cell to place your "O".

But dig a little deeper,
and it becomes clear that the model
doesn't know what is going on,
it just speaks as if it does.

It will falsely claim victory,
falsely admit defeat, or even claim a stalemate
when there is a clear winning move.

It will even play its turn on a square
that is already occupied.


J middle left

Great, you placed your "O" in the middle-left cell:

```
O|O|X
-+-+-
O|X|
-+-+-
O|  |X
```

It's my turn now. I'll place my "X" in the bottom-middle cell:

```
O|O|X
-+-+-
O|X|X
-+-+-
O|  |X
```

Unfortunately, it looks like I have won this game. But well played! Would you like to play again?

There are models that can play games.

Deep Blue beat Garry Kasparov at Chess in 1997.

AlphaGo beat Lee Sedol at Go in 2016.

But neither of these models is capable of writing essays or poetry.

Bibliography

https://www.youtube.com/watch?v=_8yVOC4ciXc

https://dev.to/ben/the-difference-between-chatgpt-and-gpt-3-19dh

https://openai.com/blog/chatgpt

https://platform.openai.com/docs/models/overview

https://jalammar.github.io/illustrated-gpt2/

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

https://jalammar.github.io/how-gpt3-works-visualizations-animations/
How GPT3 Works - Visualizations and Animations – Jay Alammar

Special thanks to
Gavin Miller
Ryan Reposar
Ian Shadforth
John Cain
Jake Sheiner

Presentation posted at
presentations.dubberly.com/generative_text_GPT.pdf