

1. Understanding Parameters and Tokens in an LLM: A Simple Breakdown

Assem Hijazi – Experience, Knowledge & Ideas

We've already established that in an LLM (like GPT), **parameters** are the relationships between words, and **tokens** "they call it tokens" are the words themselves. Now, let's dig a little deeper into how these tokens and parameters work together to help the model understand and respond to prompts.

Tokens and Relationships

First, think of **tokens** as individual **words** or parts of words. Each word has certain **relationships** with other words, and these relationships are built during the model's training. The more training data the model has, the more relationships (or **parameters**) it can establish between different words.

When you give the LLM a **prompt** (we can think of this as a "project"), the model analyzes the relationships between the words in that prompt. It **evaluates** how strongly the words are connected and decides which connections are most relevant. These relationships are measured by **vectors**, which are like numerical scores that show how related two words are. These vectors aren't static—they change dynamically based on the **context** of the prompt.

Visualizing the Relationships

Imagine each word is inside a **circle**, and from that circle, you have multiple **lines** connecting it to other words. Each line represents a potential relationship between the word in the circle and another word. Some of these lines might be **thick and strong** because those words are closely related, while others might be **thin and weak** if the relationship is less relevant.

Note: visit the site for visualize words relationship [Semantically related words for "dubai_NOUN" \(nlpl.eu\)](https://nlpl.eu/Semantically-related-words-for-dubai-NOUN)

For example, if you take the word **"bank"** and place it in a circle, you might see lines connecting it to words like **"money," "loan," "river,"** and **"finance."** The strength of these connections depends on the **context**. If you're talking about banking, the connection between **"bank"** and **"money"** will be strong, but if the context is about rivers, the connection between **"bank"** and **"river"** will become stronger instead.

Changing Contexts and Relationships

The beauty of an LLM is that these relationships **adapt** depending on the context of your prompt. Let's say you move from talking about **healthcare** to **finance** and then to **geography**:

- In **healthcare**, the word **"apple"** might be linked closely to words like **"nutrition"** and **"diet."**
- In **finance**, the word **"bank"** might be linked closely to words like **"loan"** and **"interest."**
- In **geography**, **"bank"** might shift its connection to **"river"** and **"water."**

This means that for each new context or domain, the LLM dynamically adjusts the **vectors** (relationships between words) to match the meaning that's most relevant.

The Role of Training Data in Building These Relationships

So, how does the LLM know to adjust these connections? This is where **training data** comes in. During training, the model is exposed to a huge amount of text across different topics and domains. It learns the **patterns** in language and builds initial vectors (or relationships) between words.

At the start, these vectors are **generic** and might even be random. But as the model processes more data, it refines these relationships. For instance, the more the model reads about **banks** and **finance**, the stronger the connection between **"bank"** and **"loan"** becomes.

Once trained, the model doesn't need to go back to the original data every time it gets a new prompt. Instead, it uses what it already learned—just like how you don't need to look up a fact in a book once you've memorized it. The LLM adjusts the relationships between words on the fly, based on the context of the question or prompt.

In Summary

- **Tokens** are words, and the **relationships** between those words are called **parameters**.
- These relationships are represented by **vectors**, which measure how closely two words are related.
- Depending on the **context** of the prompt, the model will adjust the strength of these relationships.
- The **training data** helps the model learn these relationships, so it can respond to different prompts in various domains (like healthcare, finance, or geography) without needing to revisit the original data.

In short, every word is connected to many other words, and the strength of those connections shifts depending on the project or prompt you're working on. It's like having a **web of words**, and the model knows how to highlight the right connections depending on what you ask.

Going a Bit Deeper: Vector Representation

If you're curious about how these relationships are actually represented in the LLM, here's a bit more detail. In models like GPT, every word (or token) is represented by a **vector**—a long list of numbers. These vectors are usually **1,024 dimensions long**, meaning each word is associated with 1,024 different numbers that capture its meaning.

For example:

- **"apple"** might have a vector like **[0.45, -0.32, 0.78, ..., -0.11]**.
- **"fruit"** could have a vector like **[0.42, -0.30, 0.80, ..., -0.10]**. Since **"apple"** and **"fruit"** are related, their vectors will be similar. In this case, the difference between their vectors is small, which shows that the LLM recognizes them as closely related.

These numbers typically range between **-1 and 1** and reflect the word's relationship with other words in the model. Words with similar meanings will have vectors that are closer together in this **high-dimensional space**, while unrelated words will have vectors that are farther apart. The model uses these vectors to understand and adjust the relationships between words based on the context of your prompt.

So, while we visualize these connections as **lines between words in circles**, under the hood, the model is using complex **mathematical relationships** between these **1,024-dimensional vectors** to decide how words are related.

This allows the model to adjust dynamically and provide relevant answers, no matter the domain or topic