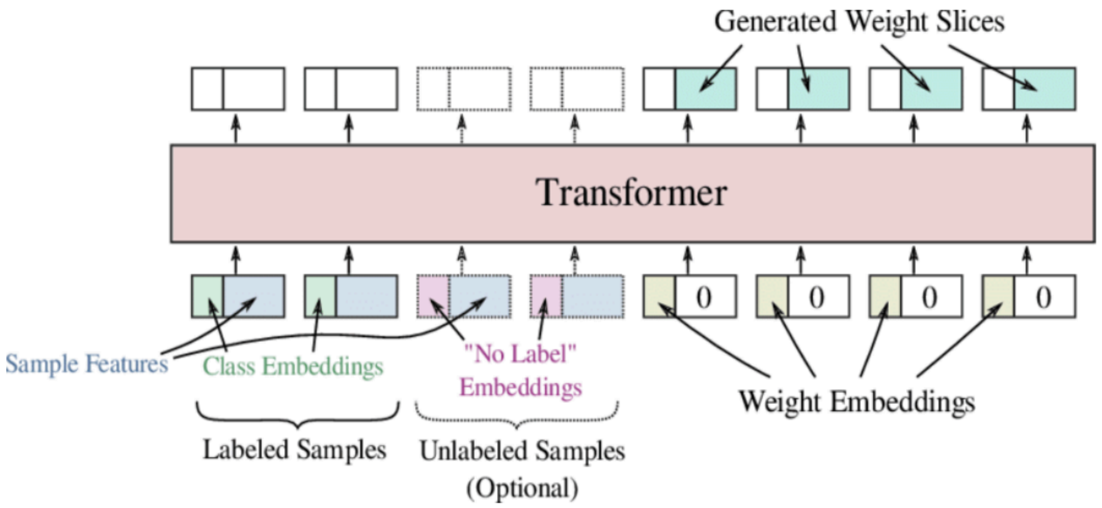
What’s a Transformer?

A **transformer** is a type of **neural network**, which is a computer model inspired by how brains work. It’s mainly used to understand or generate text, but it can also work on images, code, and more.

It has a special trick: **attention.**

The Key Pieces of a Transformer



1. Embeddings – Turning Words into Numbers

- Each word is converted into a **vector** (a list of numbers). For example:
  - “cat” → [0.2, 0.7, -0.3, …]
- These numbers capture the meaning of the word — like how “cat” and “kitten” are more similar than “cat” and “car.”

2. Positional Encoding – Adding Word Order

- Unlike sentences we read left to right, transformers look at all the words at once (like reading a sentence all at the same time).
- To help it understand **order**, we add **positional encodings** — extra numbers that tell the model the position of each word in the sentence.

3. Self-Attention – Focusing on What Matters

This is the **core** of the transformer.

Here’s what it does:

Let’s say the model is reading this sentence: “The lion saw the zebra, and it ran away.”

To understand “it ran,” the model uses **self-attention** to figure out what **“it”** is.

How?

Every word gets turned into three things:

- **Query (Q)**
- **Key (K)**
- **Value (V)**

Then it does this:

- Compare the **query** of a word with the **keys** of all other words.
- This gives a score — how important each other word is.
- Use those scores to **mix together the values**.
- That gives a new, smarter version of the word’s meaning.

So “it” might look mostly at “lion” and decide “lion” is the thing that ran.

4. Multi-Head Attention – Seeing Things from Different Angles

- Instead of doing attention once, transformers do it in **multiple “heads”**.
- Each head focuses on different types of relationships — like grammar, meaning, etc.
- Then they combine the results to get a richer understanding.

5. Feedforward Network

- After attention, each word’s vector goes through a little neural network (just a few layers of math).
- This helps the model learn more abstract patterns.

6. Layer Normalization + Residual Connections

- These help the model train better:

    - **Residual**: Adds the original input back in (like a shortcut).
    - **Normalization**: Keeps values stable so learning doesn't go crazy.

7. Stacking Layers

- All of that (attention + feedforward) is **repeated multiple times** — often 12, 24, or even 96 layers.
- Each layer builds a deeper understanding.

Final Output

If you're using the transformer to:

- **Understand** something (like in BERT), you use the final vectors for each word.
- **Generate** something (like in GPT), the model turns the final output into the next word using a softmax (a way to pick the most likely next word).

Summary:

A transformer turns words into vectors, adds info about their position, and uses **self-attention** to figure out which words are important to each other. Then it passes everything through layers of math to learn complex patterns in the data — and that's how it can answer questions, write essays, or translate languages.