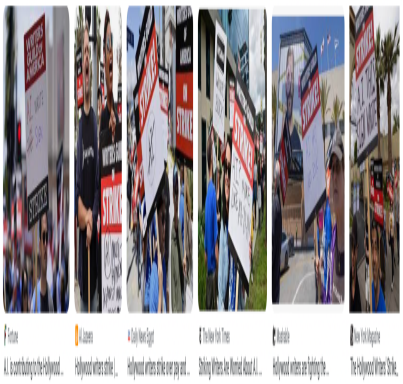

More on automated recollection

Description

Conversational AI platforms such as ChatGPT generate predictions for what should come next after a user inputs a question, statement, paragraph, or other text prompt.

In early text prediction software, a simplistic language model might calculate the most likely word to follow a given input like â??door,â?• based on pre-calculated statistical analysis of word co-occurrences in many example sentences, leading to predictions like â??hingeâ?• or â??handle.â?•

GPT models are much more sophisticated. The input to the pre-trained neural network is not simply a vector associated with the most recently added token (e.g., â??doorâ?•). Instead, the GPT model evaluates what is likely to follow â??doorâ?• in the context of a sequence of tokens. It does this by utilizing a neural network that has been trained on thousands of contexts where â??doorâ?• and other words occur.

When a user enters a sequence of words and submits it, the platform processes the entire input, which includes the userâ??s previous inputs and the modelâ??s previous responses from the same session. As the conversation progresses, the amount of information to process grows. However, this process is repeated from scratch each time a response is generated, not continuously updated in real time.

To generate a response, the platform performs calculations such as determining attention scores for individual tokens, creating key, query, and value vectors for each token, and handling the positional encoding of each token in the sequence. It also accounts for various potential outcomes through its parallelized multi-head attention procedure.

Each token in the sequence is ultimately associated with a new vector (of dimensionality corresponding to the specific model architecture, e.g., 1024 floating point numbers in GPT-3) that encapsulates its contextual information. This vector is then fed into the pre-trained neural network to compute a probability distribution over all possible next tokens in the vocabulary. The next token in the modelâ??s output is selected based on this distribution, typically with some degree of randomness to ensure diverse and natural-sounding responses.

The computational load for the ChatGPT model indeed increases as the conversation progresses. With every new interaction, the context for each token subtly shifts, requiring the model to recalibrate its parameters. This process is repeated for each interaction until the conversation reaches a set limitâ??about 25,000 tokens for the current iteration, ChatGPT4. The conversation may continue beyond this point, but the model will â??forgetâ?• the earlier exchanges that exceed this limit.

To visualize this, consider that the model processes the conversation as if it were a single, extensive sentence that could consist of hundreds of words. These words include not just the most recent user input, but also the prior parts of the ongoing conversation, up to the 25,000-token limit.

Despite the significant computational load, ChatGPTâ??s responses to user inputs are impressively fast. This speed is largely enabled by specialized server hardware, such as Graphics Processing Units (GPUs), which are designed to perform matrix operations quickly and in parallel. The transformer model used for inference is also optimized for these types of calculations. Furthermore, the modelâ??s performance during inference is largely attributable to the pre-trained neural network. This network transforms a vector representing a token into a prediction for the next token in the sequence, with the input values propagated rapidly through the layers of the network.

Beyond raw computational power and an efficient model architecture, however, the speed of ChatGPTâ??s responses also relies on various engineering optimizations implemented during both training and inference stages. For instance, strategies like smart batching and pruning can help manage the computational load and hasten response times.

As testament to the speed and competence of ChatGPT4, I wrote this post and asked the platform to evaluate it. I delivered my text in two separate exchanges. After each response from the platform I asked it to rewrite my text to implement the changes. The corrected text is disarmingly better than what I wrote originally! See exchange as a PDF. Striking script writers have a point: See Holdsworth, Lisa. (2023). â??We write the TV, radio and theatre shows you love. Do you want robots and AI doing that job?â?• *Guardian* 14 June. Retrieved 16 July 2023, from https://www.theguardian.com/commentisfree/2023/jun/14/screenwriters-world-us-strikers.

Category

1. Artificial Intelligence

Tags

1. memory
2. natural language processing

Date Created
August 5, 2023
Author
rcoyne99