Text Generation with Temperature and Top-p Sampling in GPT Models: An In-Depth Guide

Dr. Prakash Selvakumar
NLP Data Science Leader - Client Solutions and Product Innovation
Published Apr 29, 2023

+ Follow

Generative Pre-trained Transformers (GPT) are a class of powerful language models that have revolutionized the field of natural language processing (NLP). GPT models are capable of various tasks, such as text completion, translation, and question-answering, thanks to their ability to generate coherent and contextually relevant text. In this article, we will explore the concepts of temperature and top-p sampling in GPT models, illustrating their importance in generating diverse and high-quality text outputs.

Temperature and Top-p sampling are two essential parameters that can be tweaked to control the output of GPT models used in various applications like chatbots, content generation, and virtual assistants. As a business user or functional professional, understanding these parameters can help you get the most relevant responses from GPT models without needing extensive data science knowledge.

1. Temperature: This parameter determines the creativity and diversity of the text generated by the GPT model. A higher temperature value (e.g., 1.5) leads to more diverse and creative text, while a lower value (e.g., 0.5) results in more focused and deterministic text.

2. Top-p Sampling: This parameter maintains a balance between diversity and high-probability words by selecting tokens from the top-p most probable tokens whose collective probability mass is greater than or equal to a threshold p. It helps ensure that the output is both diverse and relevant to the given context.

When and How to Tweak These Parameters

As a business user, you might need to tweak these parameters to get the desired output quality, depending on the specific use case.

Temperature:

- If the generated text is too random and lacks coherence, consider lowering the temperature value.
- If the generated text is too focused and repetitive, consider increasing the temperature value.

Top-p Sampling:

- If the generated text is too narrow in scope and lacks diversity, consider increasing the probability threshold (p).
- If the generated text is too diverse and includes irrelevant words, consider decreasing the probability threshold (p).

As a business user, you can start with default values and then adjust them based on the quality of the generated text and the specific requirements of your application. It is essential to use these parameters wisely to get the most relevant responses from GPT models.

Lets get into mathematical details:

The Role of Temperature in GPT Models

Temperature is a hyperparameter that affects the probability distribution of the tokens generated by the GPT model. By adjusting the temperature, we can control the diversity and creativity of the generated text. Higher temperature values result in more diverse outputs, while lower values lead to more focused and deterministic text.

Mathematically, the temperature is incorporated into the softmax function, which is used to convert the logits (raw output scores) produced by the GPT model into probabilities:

$$P(\text{token}_i) = \exp(\text{logits}(\text{token}_i) / T) / \Sigma_j \exp(\text{logits}(\text{token}_j) / T)$$

Where:

- P(token_i) is the probability of generating token i
- logits(token_i) is the logit (raw output score) for token i
- T is the temperature
- Σ_j is the sum over all tokens in the vocabulary

By varying the temperature, we can strike a balance between the diversity and the quality of the generated text, depending on the specific use case.

Understanding Top-p Sampling in GPT Models

Top-p sampling, also known as nucleus sampling, is an alternative to conventional sampling methods that involve selecting the most probable tokens. Top-p sampling helps generate more diverse and creative text by considering a broader range of tokens.

In top-p sampling, a probability threshold p is chosen, and tokens are sampled only from the top-p most probable tokens that collectively have a probability mass greater than or equal to p. This ensures that the model considers a broader range of possible tokens while still prioritizing higher probability ones.

Practical Examples and Workings of Temperature and Top-p Sampling

To illustrate the concepts of temperature and top-p sampling, let's consider a GPT model tasked with generating the next word in a sentence based on the given context. We will explore how different temperature settings affect the generated word probabilities. For simplicity, we assume the model's vocabulary consists of six words: achieve, conquer, dream, succeed, the, and world.

Assuming the raw output scores (logits) from the GPT model are as follows:

achieve: 3

conquer: 1

dream: 1

succeed: 2

the: 4

world: 2

Low temperature (T=0.5):

First, we will adjust the logits using the low temperature value:

achieve: 3/0.5 = 6

conquer: 1/0.5 = 2

dream: 1/0.5 = 2

succeed: 2/0.5 = 4

the: 4/0.5 = 8

world: 2/0.5 = 4

Applying the softmax function at low temperature:

Sum of exponentials: exp(6) + exp(2) + exp(2) + exp(4) + exp(8) + exp(4) ≈ 7384.56

Probabilities:

achieve: exp(6) / 7384.56 ≈ 0.121

conquer: exp(2) / 7384.56 ≈ 0.004

dream: exp(2) / 7384.56 ≈ 0.004

succeed: exp(4) / 7384.56 ≈ 0.032

the: exp(8) / 7384.56 ≈ 0.810

world: exp(4) / 7384.56 ≈ 0.032

With a low temperature setting, the probabilities become more focused on the highest probability tokens. In this example, the word "the" has a significantly higher probability than the other words, making the generated text more deterministic and

less diverse.

High temperature (T=1.5):

Now, we will adjust the logits using the high temperature value:

achieve: 3/1.5 = 2

conquer: 1/1.5 ≈ 0.67

dream: 1/1.5 ≈ 0.67

succeed: 2/1.5 ≈ 1.33

the: 4/1.5 ≈ 2.67

world: 2/1.5 ≈ 1.33

Applying the softmax function at high temperature:

Sum of exponentials: exp(2) + exp(0.67) + exp(0.67) + exp(1.33) + exp(2.67) + exp(1.33) ≈ 18.20

Probabilities:

achieve: exp(2) / 18.20 ≈ 0.263

conquer: exp(0.67) / 18.20 ≈ 0.106

dream: exp(0.67) / 18.20 ≈ 0.106

succeed: exp(1.33) / 18.20 ≈ 0.184

the: exp(2.67) / 18.20 ≈ 0.303

world: exp(1.33) / 18.20 ≈ 0.184

With a high temperature setting, the probabilities are more evenly distributed, resulting in a more diverse and creative selection of words. In this example, the word "the" has the highest probability, followed by "achieve." However, the other words still have a significant chance of being selected, allowing for more varied sentence completion.

Top-p Sampling:

For our top-p sampling example, let's assume the GPT model generates the following probabilities for each word in the vocabulary:

achieve: 0.05

conquer: 0.10

dream: 0.35

succeed: 0.15

the: 0.25

world: 0.10

Assuming a probability threshold of p=0.6, we will select the top-p most probable tokens that collectively have a probability mass greater than or equal to 0.6.

Sorted probabilities:

dream: 0.35

the: 0.25

succeed: 0.15

The top-p tokens are "dream," "the," and "succeed," which collectively have a probability mass of 0.35 + 0.25 + 0.15 = 0.75, satisfying the threshold. The GPT model will then sample the next word from these three tokens, maintaining a balance between diversity and high-probability words.

In **conclusion**, temperature and top-p sampling are essential tools for controlling the diversity and quality of text generated by GPT models. By understanding and leveraging these concepts, we can harness the full potential of GPT models in a wide array of NLP tasks and applications.

-GPT assisted article