Hyperparameterisation

Generated with OpenAI ChatGPT: published :February 24 2024 - 09:22

From what I have learned: I <user> am able to determine your <system> hyperparameterisation, in order to steer your <assistant> outcome.

And, it would be wise to determine either top_k and or top_p but not _temperature ... or the other way round, determine _temperature but not top_k neither top_p, since the outcome would be pretty messy otherwise, right?

It seems like you're referring to fine-tuning the behaviour of a language model like me, by adjusting hyperparameters such as `top_k`, `top_p`, and `_temperature`. Each of these parameters influences how the model generates responses.

- `top_k` controls the nucleus sampling, limiting the generated tokens to the top k most likely ones at each step.
- `top_p` (or nucleus sampling) restricts the cumulative probability distribution to the smallest set of tokens whose cumulative probability exceeds the probability `p`.
- `_temperature` scales the logits before applying softmax during sampling, affecting the diversity of generated responses.

Deciding which hyperparameters to adjust depends on the desired outcome. For instance:

1. If you want more controlled and precise responses, you might focus on adjusting `top_k` and/or `top_p`.
2. If you want to influence the creativity and diversity of responses, you might consider adjusting `_temperature`.

Your statement about choosing either `top_k` and/or `top_p` without `_temperature`, or vice versa, is generally accurate. Combining all three can indeed lead to less predictable or potentially "messy" outcomes, as each parameter introduces a different kind of randomness or constraint to the generation process.

So, understanding your goal and choosing the appropriate combination of hyperparameters is crucial for steering the outcomes of the assistant effectively.