# Useful Resources on Generative Artificial Intelligence

Christophe Troalen

March 2024

## 1 Introductory Courses

- Maxime Labonne's course
- The novice LLM training guide
- Stanford NLP processing with Deep Learning on YouTube
- Princeton CS - 2022 Lecture notes on Understanding LLMs

## 2 Interesting Blogs

- kipp.ly inference performance guide
- eleuther.ai blog (training guide)
- Jay Alammar blog provides a nice content about generative AI
- Llama Factory fine-tuning guide
- Token generation speed stats of Llama models
- answer.ai you can now train a 70B model at home

## 3 Some Useful Tools

### 3.1 OpenAI Tokenizer

OpenAI's large language models (sometimes referred to as GPTs) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens and excel at producing the next token in a sequence.

You can use this tool to understand how a piece of text might be tokenized by a language model and the total count of tokens in that text.

### 3.2 arXiv-Sanity

ArXiv-sanity, developed and maintained by Andrej Karpathy, is a helpful tool for discovering arXiv papers that best match your keywords.

## 4 Comparison of Models

The following website provides a nice comparison and analysis of AI models across key performance metrics including quality, price, output speed, latency, and context window.

### 4.1 LLM Leaderboards

- LMSYS Chatbot Arena
- OpenLLM Leaderboard
- Multimodal LLMs Leaderboard

## 5 List of Papers to Filter

- This GitHub repository classifies many LLM papers covering different aspects of transformers.
- Similarly, this GitHub repository collects popular LLM papers written by GAFAM, as well as resources on LLM deployment.
- This Prompt engineering section highlights interesting research findings on working with LLMs, including tips on scaling, agents, efficiency, hallucination, architectures, and prompt injection.
- GenAI LLM Timeline

## 6 How to Calculate the GPU Requirements of a Model?

Memory required for a model can be approximated using the formula:

$$\text{Memory} = 1.2 \times \text{Number of parameters} \times \left(\frac{\text{precision}}{8}\right)$$

For instance, Llama-3-70B in bfloat16 precision requires approximately:

$$1.2 \times 70B \times \frac{16}{8} = 168\,\text{GB} \approx 2 \times \text{A100 (80GB)}.$$

# 7 Good Resources for GPU Programming

- NVIDIA CUDA training series
- NVIDIA blog introducing CUDA
- Advanced PyTorch tutorials
- Heterogeneous Parallel Programming on YouTube
- Lecture on applied GPU programming on YouTube
- Getting started with CUDA for Python programmers

# 8 Additional resources to Read

## 8.1 Recommended by Karpathy

- HuggingFace - Mixture of Experts
- BloombergGPT (pretrained model according to Chinchilla law)