What I Learned Building a Mini-GPT Transformer Model

Alexis Ambriz
3 min read · Just now

I built a mini or nano generative pre-trained transformer model clone using google colab! You can view the notebook I used here. I followed this colab notebook as guidance, however, I also used my own training data and had to make some modifications to the original colab to get the model running.

Extracting, Loading, and Transforming the Ebook Training Data

I chose the book “Voices from within the Veil” by W.E.B DuBois. This book was taken from the Project Gutenberg website and is available to use freely. I also would like to read this book on my own spare time! I uploaded it to Google Drive for my own use as to not scrape Project Gutenberg’s website every time I reran the notebook during debugging. The specific debugging steps I took was making sure that the hyperlink to the data was constructed properly to avoid running the notebook on empty data. I also had to debug the tokenization schema by making my own tokenizer.

This tokenizer was used on the training data to avoid training the model on tokens that are outside the vocabulary range of the model. This could happen for instance if you are using a tokenizer that is purpose-made for GPT-4 or other foundation models. Since they are trained on larger datasets, I found there were issues using their tokens to encode my own data.

One thing to note is that as this is a 'Nano GPT', the data is small and is around ~419 KB. Given this, the transformer model architecture may overfit to the data and return nonsense output if given a input context that is not captured exactly within the training data.

Initializing the Mini Transformer Model

The model architecture for the mini transformer model encapsulates many of the key elements that used to build a full transformer model. The model uses Pytorch neural network classes and as such it is possible to run this model on a GPU environment within colab or on your own local devices. The specific classes that were built as part of the nano-GPT architecture includes: the head, multi-head attention, feed forward, and block classes.

Of these classes, the head and multi-head attention classes are unique to the transformer/GPT architectures. They are responsible for encoding the input tokens and tracking their influence on other words that are within the input, whether they are in a earlier or later index or location in the input context. This allows the GPT architectures to understand complex patterns that arise in natural language and allow for the development of Large Language Models. The feed forward and block classes are more typical and common in other architectures as they define neural network models in general and allow them to process information, update their knowledge, and make predictions.

Training the Model & Generating Text Completions

After initializing the model and ensuring the data was being properly encoded and decoded using the tokenizer, it was time to train and evaluate the model. This is also when it is necessary to evaluate the hyper parameters used for training, such as the learning rate and training steps. Initially, the training steps were set to 5000 when I was using the gpt-2 tokenizer and I noticed I was able to lower the loss for training to ~0.5. However, there were difficulties applying this tokenizer as many of the tokens that were being predicted post training lied outside of the range of the tokenizer and this may be an issue resulting from improper training or insufficient data or model architecture. Using the custom tokenizer allowed for applying a constraint to the tokenizer's vocabular of available tokens to prevent this issue. It also lead to a higher loss for some reason, thus I had to increase the training steps to 10000. The model was still only able to reach a training loss of ~1.5 and validation loss of ~1.9.

To evaluate the model, I used the input of the ebook's title 'Voices from within the ____' — I assume this is a good test case as it should've appeared multiple times within the ebook text. The expected output only requires the model to evaluate one word ('veil'). However, the model output 'Voices from within the givered ramily line'. If I had to guess, I believe the model may have filled in with 3 words as opposed to 1 because the 'max_new_tokens' parameter was set to 20 — the exact number of characters present within the new words generated outside of the input context. I believe allowing for the max_new_token parameter to be a variable value would allow for the model to make better predictions.

Data Science Software Development

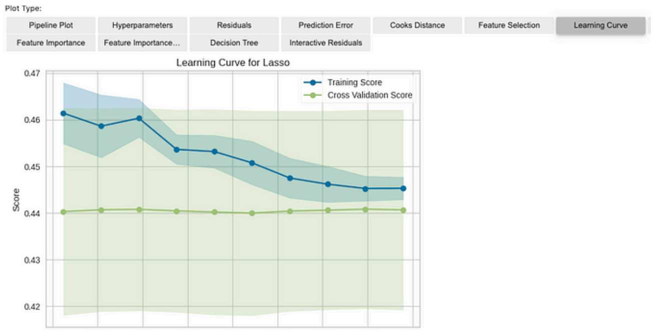
Written by Alexis Ambriz

0 Followers · 9 Following

Edit profile

SWE & Data Analyst

More from Alexis Ambriz



Alexis Ambriz

An epidemiological analysis of Sars-CoV-2 using PyCaret

In this article, I will guide you through an analysis of Coronavirus / Sars-CoV-2 data…

Sep 20, 2023 66

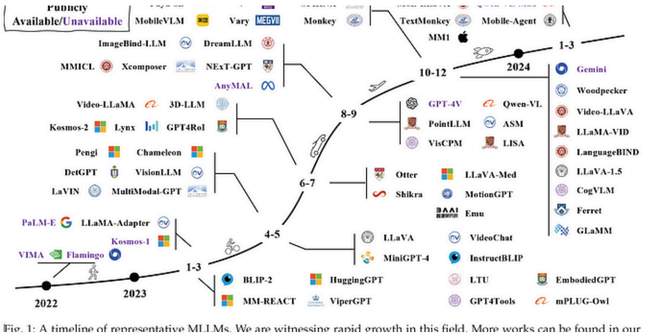


Fig. 1: A timeline of representative MLLMs. We are witnessing rapid growth in this field. More works can be found in our

Alexis Ambriz

Review of a Survey of Contemporary Multi-modal A.I…

What makes them special? Let's find out.

3d ago

Alexis Ambriz

A Guide to Building a Full-Stack Live Audio Chat Room App

Introduction

Aug 23 2

Alexis Ambriz

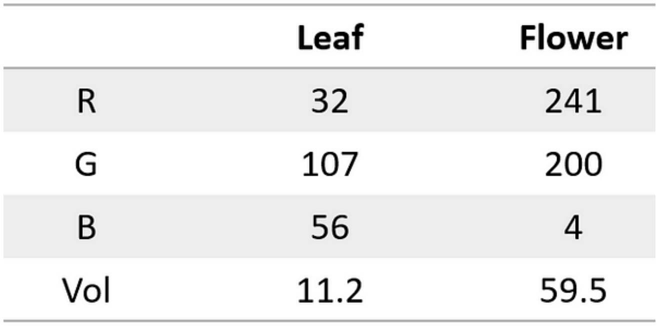
Revolutionizing Wireless Communication: Machine Learnin…

Imagine you're streaming your favorite show, video chatting with a friend across the globe,…

Nov 7, 2023 11

See all from Alexis Ambriz

Recommended from Medium



	Leaf	Flower
R	32	241
G	107	200
B	56	4
Vol	11.2	59.5

In Towards Data Science by Rohit Patel

Understanding LLMs from Scratch Using Middle School Math

In this article, we talk about how LLMs work, from scratch—assuming only that you know…

Oct 19 👏 5.1K 💬 68

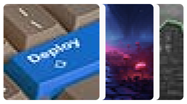
Jesper Nordström

Multi-Agent Systems: A Transformative Paradigm in AI

One of the most promising and rapidly evolving areas within AI right now is the real…

✦ Jun 16 👏 222 💬 2

Lists



Predictive Modeling w/ Python

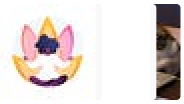
20 stories · 1681 saves

Coding & Development

11 stories · 911 saves

General Coding Knowledge

20 stories · 1761 saves



Stories to Help You Grow as a Software Developer

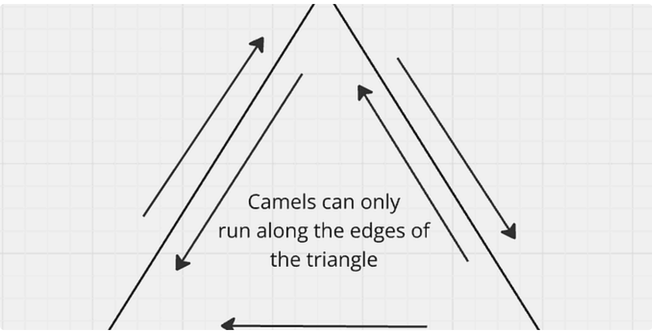
19 stories · 1485 saves

In Accredian by Vinay Dhurwe

Harnessing Unsupervised Learning for Anomaly Detection in Legal…

Exploring AI's Role in Transforming Legal Analysis by Uncovering Hidden Patterns and…

3d ago 11



Lucas Samba

3 Probability Questions I was asked in Walmart Data Scientist Interview

Recently I got an opportunity to interview at Walmart for Data Scientist—3 position. All…

✦ Aug 22 973 27

In Funny, Inc. by Murphy's Law

This LinkedIn Post Got Me 12 Job Offers

Hacking the system starts at home

✦ Aug 10 6.1K 143

Harendra

How I Am Using a Lifetime 100% Free Server

Get a server with 24 GB RAM + 4 CPU + 200 GB Storage + Always Free

✦ Oct 25 5.3K 70

See more recommendations