LLM-MRI Python module: a brain scanner for LLMs

Luiz Costa[1], **Mateus Figenio**[2], **André Santanchè** [1], **Luiz Gomes-Jr** [2]

[1]Instituto de Computação – UNICAMP – Campinas, SP – Brasil

[2]Departamento de Informática – UTFPR – Curitiba, PR – Brasil

santanch@unicamp.br, lcjunior@utfpr.edu.br

Abstract. *LLMs (Large Language Models) have demonstrated human-level language and knowledge acquisition skills in several tasks. However, despite the recent success and broad use, understanding how these skills are learned and encoded inside the underlying neural network is still challenging. The goal of the LLM-MRI package is to simplify the study of activation patterns in any transformer-based LLM, similarly to how MRI (magnetic resonance imaging) simplifies with biological brains. The package, written for the Python language, allows the mapping of neural regions using a parameterized reduction of the model's dimensionality. Neural regions can be visualized according to the forward-pass activations stimulated by a set of documents. Similarly, the package enables the creation of graph models representing the interlayer network of connections stimulated by a set of documents. These features allow for qualitative and quantitative assessments of the underlying structure of activations, depending on the type of documents that the LLM model is exposed to.*

1. Introduction

The field of Natural Language Processing (NLP), an area of research that aims to enable computers to process large datasets of natural language, has been revolutionized since the introduction of language models based on deep learning approaches. LLMs (Large Language Models) have been in the spotlight with their surprising discursive quality and pervasive use in many day-to-day applications [Hiter 2024].

However, our understanding of how language models based on neural network approaches operate has lagged behind the evolution of their architecture, increase in size, expansion of training databases, and their broader use across society. Models based on neural networks are notoriously considered "black box" models, given their multi-layer structure and non-linear relationship between input and output values. To address this problem, the Explainable Artificial Intelligence (ExAI) research field has been developing methodologies and tools that seek to better understand and explain how systems that rely on this mechanism develop their conclusions.

The explainability of systems that process language faces different challenges given the nature of its object of work, which involves different levels of meaning, from syntactic to semantic. Meaningful explanations benefit these systems' development by helping to validate that a system works as intended, help improve the system by better understanding its shortcomings, lead to new discoveries for experts in its field of application, and ensure that the system complies with legislation [Samek et al. 2017].

The LLM-MRI package aims to simplify the study of activation patterns in transformer-based LLMs. The package, written for the Python language, allows the mapping of neural regions using a parameterized reduction of the model's dimensionality. By analyzing the patterns of activation incited in the model for each category of text, users can study how these patterns are related to common properties of the texts in these categories and can analyze which regions are more frequently activated by each category. This provides insight into the particular features of the knowledge expressed in the documents that the neural network has learned, as well as into how the models behave for different content.

The main features offered are:

- **Dimensionality Reduction**: LLM models have high dimensionality, hindering analysis of underlying patterns. The module provides user-customizable dimensionality reduction, simplifying visualization according to the application needs. Individual neurons are mapped into regions that aggregate neurons with similar activation patterns;
- **Visual Representation of Activation**: The module creates a visual representation of the activation patterns for a given layer (user-defined), and throughout layers. The module allows the user to filter activations by document category (i.e. label);
- **Graph Representation of Activations**: The module connects regions from neighboring layers, based on co-activations, to form a graph representing the patterns of activations of the entire network. The graph can be used to extract other properties using metrics from Complex Networks, such as centrality and clustering;
- **Openness and documentation**: The package is open-source (under the license LGPL v3), available on GitHub[1] and installable through PyPI. The repository contains the documentation and usage examples. The project also aims at engaging the community in the development of the library.

The module allows users to explore questions such as: (i) do different categories of text in my corpus activate different neural regions? (ii) what are the differences in the properties of the underlying graph formed by activations of texts from two distinct categories? (iii) are there regions of activation in the model more related to specific aspects of a category?

The remainder of this research paper is organized as follows: Section 2 presents an overview of related works to the project; Section 3 describes the developed toolkit with a user example and details the artifacts it produces and its functionalities; and Section 4 concludes the paper resuming its contributions and presenting next steps of development and research.

2. Related Work

Language Models represent a probability distribution of words in natural language texts [Bengio et al. 2000] for interpreting and generating free texts. These models have been directly benefited by the enhancement of neural networks' capabilities, as they can extract semantic information from texts in a way that a machine can analyze and manipulate it [Tunstall et al. 2022]. Consequently, with the progression of this technology's

[1]https://github.com/explic-ai/LLM-MRI

development, Large Language Models (LLMs) were conceived. LLMs are models that, having been trained with substantial volumes of text and counting on a larger number of internal parameters, can approximate human performance in multiple activities [Naveed et al. 2024]. These models have considerable processing capacity, adapted to handle massive volumes of data, enabling them to interpret and produce responses to complex requests.

The main approaches to explain and interpret the inner representations of LLMs are divided into visualization and analysis approaches. The first pertains to any method through which one can visualize how the model has computed a given input or how it is structured, and the second to mathematical or statistical ways to understand the inner workings of the models.

Visual explainability approaches focused on understating how the internal representations of the model's process inputs have been extensively based on visualizing its attention mechanisms, ranging from visualizing how individual attention heads evaluate tokens [Vaswani et al. 2017], to how attention values flow across the model [DeRose et al. 2020], and how individual attention heads relate to concepts given by the user [Hoover et al. 2019]. There are also visualization tools and resources focused on explaining the models' architecture, such as LLM Visualization[2], by Brendan Bycrof, that lays out the components of the models and explains its functions.

In contrast to the cited approaches, we focus on activations of the feed-forward layers. Although attention is an important and defining aspect of transformers, we posit that activations carry more semantic meaning, as supported by [Dalvi et al. 2019] and [Geva et al. 2022], and are more amenable to our neural area mapping and graph construction approach, as described in the next section. Furthermore, the feed-forward layers already have the output of the attention heads integrated by the processing pipeline.

3. Architecture

The workings of the module are divided into three stages: Activation Extraction, Neural Region Mapping, and Artifact Output, as shown in Figure 1. These stages are detailed in the following section.

To demonstrate the use of the module, we will explore a scenario with a corpus containing news articles in two categories: Fake and True. The goal would be to understand whether patterns in the activations of true news differ from fake news. This and other examples are available in the project's GitHub repository.

For the **Activation Extraction** phase, the LLM model to be visualized is loaded (step 1). In our example, a BERT model can be defined in this step. The user also specifies the corpus to be used as a reference for the baseline activations (step 2). In our example, the corpus contains news documents. At this point, each document in the provided corpus is fed to the model (step 3). The model's forward pass generates the activations associated with each document. The activations are stored to be used in the next steps.

After processing the corpus, the **Neural Region Mapping** is started, with the library reducing the dimensionality of each layer of activations, producing a two-dimensional matrix of activations (step 4). We currently use UMAP for dimensionality

[2]https://bbycroft.net/

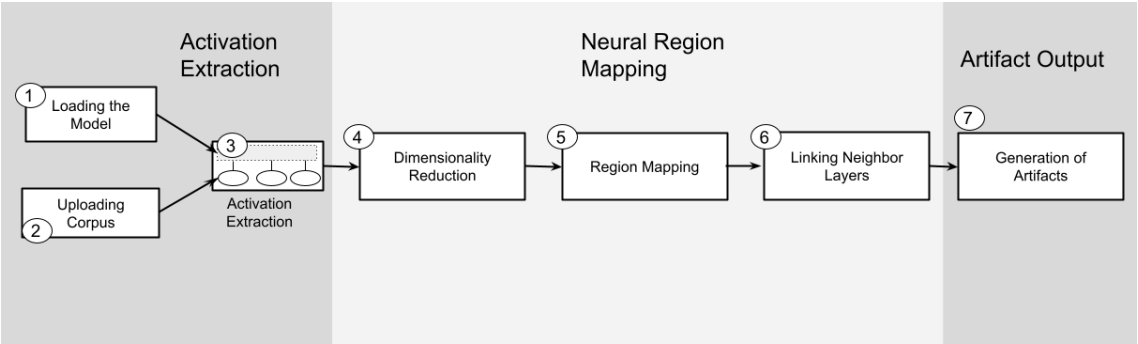

Figure 1. Diagram representing the library's functionality pipeline

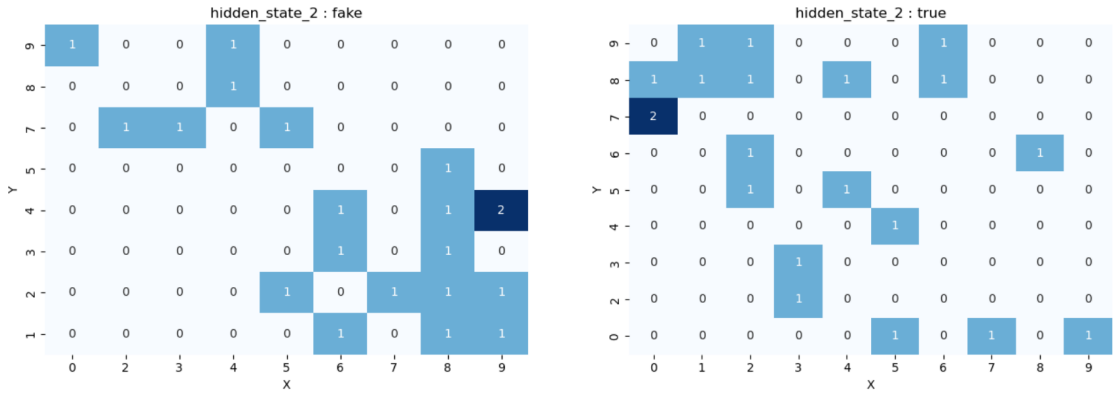

Figure 2. Visualization of activations of the second hidden state layer

reduction. After the dimensionality reduction, the two new dimensions (for each layer) are divided into 2D maps of size chosen by the user (step 5). For example, a layer with 768 neurons could be projected to maps of 10 by 10 cells. Each cell represents a region of the original higher dimensionality, i.e., an activation value in the original layer is mapped to a region of the reduced space. Figure 2 shows a mapping for the classes "Fake" and "True" with respective activations for the second layer in a 10 by 10 map.

In the final step of this phase, the underlying graph representing the activations for the entire model is created (step 6). The nodes of the graph are the neural regions in the 2D maps created in the previous steps. The edges represent subsequent activations in neighbor layers (now represented by the maps). An edge counter registers the total number of documents that activated each pair of neighbor regions. These counters are implemented as weights of the edges and can be used to trim the graph according to user preference (e.g. keeping only the strongest connections).

The last phase is the **Artifact Output**. The module currently provides three types of artifacts:

- Layer activation images, showing the 2D maps with activated regions according to the category of documents. Figure 2 shows an example of this type of image for the categories "Fake" and "True".
- Graph model, with the structure of the activations. The generated graph is a *Net-*

workx[3] model, which enables the use of the *Networkx* library for further processing, such as transformations and processing of network science metrics (e.g., clustering, centrality, etc.).

- Graph images, showing the graph representation of the activations. The user can choose to highlight the edges activated for a given type of category. Figure 3 shows a graph with two different categories highlighted, "false" in blue, "true" in orange, and green being the edges shared between the two categories.

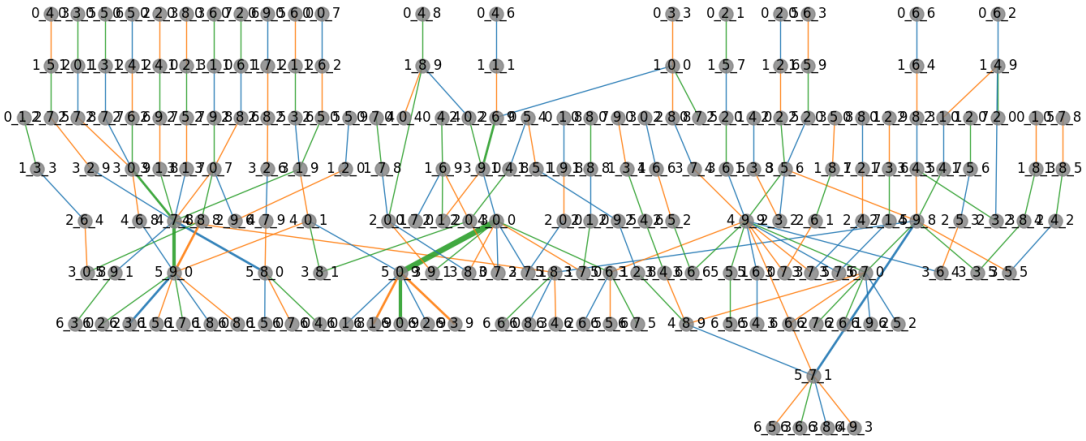

Figure 3. Zoomed in graph visualization of different types of activations through layers

4. Conclusion

This paper presents a Python module for understanding how LLMs respond to different types of texts based on the activations that the texts elicit on the model. The module offers a novel representation for analyzing LLMs by mapping the activation values of neural layers into activation regions and building a graph representing them. We expect to enable users to reach an intuitive understanding of how information and patterns in the texts are reflected in the LLM models. This type of knowledge can be used both to further understand the inner workings of LLMs and to surface differences in patterns from different documents in a corpus.

The module is in active development and use for research. Different neural network implementations are being used to verify the library's effectiveness for various types of models. Among these, the project collaborates with doctors in Brazil, the Netherlands, and China for research focused on analyzing differences in activations in texts produced by doctors with different levels of knowledge. The central question is whether activation patterns can be related to doctors' expertise concerning a given health subject.

Our next steps are to expand the library tools with more ample visualization options and offer quantitative metrics for analysis of the graphs of activation generated, based on graph theory and complex networks parameters. Future research paths are to evaluate how significant such metrics are to understand how LLMs represent different

[3]https://networkx.org/

types of texts, what kind of information from the corpus is better highlighted in its inner representations, as well as to evaluate how the mapping and graphing functions here presented can be tuned to better capture such patterns.

References

Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Dalvi, F., Durrani, N., Sajjad, H., Belinkov, Y., Bau, A., and Glass, J. (2019). What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6309–6317.

DeRose, J. F., Wang, J., and Berger, M. (2020). Attention flows: Analyzing and comparing attention mechanisms in language models.

Geva, M., Caciularu, A., Wang, K. R., and Goldberg, Y. (2022). Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space.

Hiter, S. (2024). Top 20 generative ai tools and applications in 2024. Disponível em: `https://www.eweek.com/artificial-intelligence/generative-ai-apps-tools/`.

Hoover, B., Strobelt, H., and Gehrmann, S. (2019). exbert: A visual analysis tool to explore learned representations in transformers models.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2024). A comprehensive overview of large language models.

Samek, W., Wiegand, T., and Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models.

Tunstall, L., Von Werra, L., and Wolf, T. (2022). *Natural language processing with transformers*. " O'Reilly Media, Inc.".

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.