Investigating the Localization of Mathematical Knowledge in GPT Models

Henriette Kohnen
u299066

Abstract

This paper investigates the localization and manipulation of mathematical operator knowledge in a fine-tuned GPT2-medium model. Using Causal Mediation Analysis and Rank-One Model Editing, the representation of addition, subtraction, multiplication, and division within middle-layer multilayer perceptron modules is identified. Successful edits to a fine-tuned math-model and observed generalization effects are demonstrated and support the localization hypothesis. These results highlight the potential of mechanistic interpretability techniques for understanding how large language models represent knowledge about mathematical operators.

1 Introduction and Related Work

In current times, large language models (LLMs) like OpenAI's ChatGPT have become increasingly prevalent in our society (Hosseini et al., 2023). Given the complexity of those models, understanding and interpreting them is becoming a highly interesting branch of research, and a variety of different approaches has been presented in the contemporary literature. As a recently emerging sub-field of interpretability, mechanistic interpretability methods including probing, sparse autoencoders, visualization, automated feature explanation, ablation, and causal mediation analysis (CMA) have shown promising results towards a better understanding of LLMs (Rai et al., 2024). One of the highly influential papers on the interpretability of LLMs, more specifically working with auto-regressive generative pre-trained transformer (GPT) models was published by Meng et al. (2022), who developed a workflow for the localization and editing of factual associations. This workflow included CMA to localize the neuron activations of a GPT model responsible for the model's predictions on factual prompts. Meng et al. (2022) found that the set of decisive neuron activations could indeed be identified at the site of middle-layer multilayer perceptron (MLP) modules. To further support their localization hypothesis, the authors developed a technique called Rank-One Model Editing (ROME), performing effective edits on the model weights without the need for re-training.

While factual knowledge seems to be emerging rather naturally even in earlier and smaller versions of auto-regressive GPT models (Radford et al., 2019; Brown, 2020), the mathematical abilities, which are tightly linked to more advanced reasoning tasks, have only been shown to bring more impressive results in more novel and bigger models like GPT4 (Achiam et al., 2023; Bubeck et al., 2023; Frieder et al., 2024). Previous smaller models like GPT2 (Radford et al., 2019) and even GPT3 (Brown, 2020) that have the advantage of being computationally more tractable than GPT4 were shown to be struggling with a more general arithmetic understanding (Mishra et al., 2022). Accordingly, only rather few past studies have presented approaches to explaining and understanding their mathematical abilities. For example, Zhang et al. (2023) propose a step-by-step planning solution to improve the interpretability and accuracy of GPT2's performance on math word problems. A recent study by Hanna et al. (2024) investigates how GPT2 computes the mathematical relation greater-than by identifying and explaining circuits within the model using path patching. Zhang et al. (2024) similarly propose a path patching approach to identify key components responsible for the mathematical abilities of LLaMA2-7B and LLaMA2-13B (Touvron et al., 2023), however, they move on to showing that fine tuning only those identified MLP modules can improve the models' mathematical abilities while minimizing the impact on non-mathematical computations.

Given the success of Zhang et al. (2024) using a path patching approach to narrow down the localization of the modules responsible for mathematical computations and the highly promising methodology proposed by Meng et al. (2022), the present paper aims at utilizing this methodology to investigate the way in which LLMs store mathematical knowledge. More specifically, the focus will be on the localization of the four basic mathematical operators plus, minus, times, and divided by. This paper will present a study consisting of two parts: First, a replication of the results proposed by Meng et al. (2022) using the slightly smaller GPT2-medium model will be presented as a baseline to compare against the results of the second part of the study. Secondly, the same methodology will be applied to another version of the GPT2-medium model fine-tuned to solve elementary-level math problems.

Overall, this paper will aim at answering the following research question: *Can knowledge about mathematical operators be successfully located and edited in a GPT2-like model using causal mediation analysis and rank-one model editing?*

In the following, methods will be explained before the results will be presented and discussed. It will be demonstrated that knowledge about mathematical operators can indeed be localized in early MLP modules of the network of a fine-tuned GPT2-medium model. Compared to factual associations, the impacts of CMA are even stronger, and possible reasons for this will be discussed. An exemplified application of the ROME methodology (Meng et al., 2022) will show that editing the previously localized MLP modules can indeed change the outcome of single exercises, and instances of generalization in the interpretation of single numbers and operators will be presented.

2 Methods

The methodology for the CMA and the ROME technique is taken from Meng et al. (2022) for the replication of their results and is slightly adapted to fit the mathematical application presented in this paper. The following will therefore only give a rather short overview of those two methods. For a more detailed explanation of CMA and ROME, please refer to Meng et al. (2022) directly.

Causal tracing, the CMA used by Meng et al. (2022) works in three steps: First, in the clean run, a prompt is issued into the model and all hidden activations are collected and saved. Secondly, in the corrupted run, the tokens corresponding to the part of the input, whose associations are to be investigated, are altered by adding an error term to the hidden activations immediately after embedding the respective parts. In the case of the factual associations investigated by Meng et al. (2022), the activations corresponding to the embedded subject are obfuscated, while in the present mathematical application the embeddings corresponding to the mathematical operators (plus, minus, times, and divided by) are changed. The third and crucial step of causal tracing is the corrupted-with-restoration run. Here, the model gets the same corrupted embeddings as its input, however now single clean hidden states taken from the first run are subsequently patched into the model's set of internal activations. For each run all internal activations are monitored and the probability of the model making the correct prediction is collected.

Like this, a total effect (TE) of the corrupted input embeddings on the model's output can be calculated as the difference in the probability of emitting the correct output in the clean run versus the corrupted run. The indirect effect (IE) of each single hidden activation on the model output can be calculated as the difference in the probability of emitting the correct output between the partially restored corrupted run and the fully corrupted run. Those activations which are able to recover the correct output for the model even though the other hidden states remain corrupted, are the ones that are of causal importance for retrieving the knowledge about the subject or the mathematical operator (Meng et al., 2022).

To get a better and more stable measure of those effects, the TE as well as the IE for each hidden state were averaged over a sample of 100 factual associations or math exercises respectively to get the average TE (ATE) and average IE (AIE). Of those 100, only items that were correctly predicted in the clean run were considered for the measures. Since each of the hidden states contains activations for the MLPs as well as for the attention modules, the more specific location of interest was found by observing the AIEs for restoring the whole hidden state, only the MLPs' activations, or only the attention modules' activations during the corrupted-with-restoration run. Additionally, path specific effects were examined by freezing either the MLP or the attention modules' activations in their corrupted state when patching in their state's cleaned activations. Like this, the direct effect of non-restoration of the specific modules can be observed in the changes of the AIE.

The idea of the ROME technique is based on viewing MLPs as a kind of associative memory into which new key-value pairs can be inserted relatively easily by solving a constrained least-squares problem (Meng et al., 2022). In the case that knowledge can be localized in the MLP modules of a model, adding new associations to the model can thereby be done by inserting them into the weight matrices of the identified MLPs. ROME is also performed in three steps: First, a new subject or mathematical operator is passed into the model as input and the activations at the previously identified MLP site are collected. This is the key for the new key-value pair. Secondly, the value is found by performing weight optimization using backpropagation until the wished for model output is generated and the vector output at the crucial MLP can be collected to be the new value. Third, the key-value association can be inserted into the MLP weight matrix using a rank-one update. For the mathematical details please refer to Meng et al. (2022).

One prerequisite for the causal tracing analysis is a fitting dataset: Meng et al. (2022) used a set of 1000 factual associations, while the present math application uses a mathematical benchmarking dataset SVAMP (Patel et al., 2021) as a basis for creating a dataset whose prompts can be used for the CMA. Before usage for the CMA, the dataset was restructured. Each exercise in the SVAMP was formed into a simple prompt of the format {number} {mathematical operator} {number} {equals} and another number as the prediction target and solution to the exercise, whereby only exercises with one operator were taken into consideration to simplify the causal tracing process. Like this, a total of 763 simple math exercises were created.

Another step that was taken before the start of the mathematical analysis was the fine-tuning of the GPT2-medium model. For the replication part of this study, a GPT2-medium model with 345M parameters and 24 layers (Radford et al., 2019) was used. This model was initially not able to solve the exercises provided in the novel SVAMP based math datasets, wherefore fine-tuning became necessary. Fine-tuning applications of GPT2 models can be found in the literature relatively often, for example for the automatic generation of German drama texts (Bangura et al., 2023) or patent claim generation (Lee and Hsiang, 2020). After hyperparameter optimization and evaluation on different sizes of training data, the present model was finally trained for five epochs on 15.000 mathematical exercises resembling the form of the prompts used for CMA with a learning rate of 3.15e-05 and batch size 16. After fine-tuning, the model, which will hereinafter be called GPT2-Math, predicted the target token of a mathematical exercise from an unseen test set of size 2.500 with 45.88% accuracy. Given the fact that out of 100 factual associations, the GPT2-medium model for the replication part of this study predicted 51 correctly, this accuracy was taken to be sufficient for the present experiment.

All computations were run using Google Colab's T4- or A100-Nvidia-GPU.

3 Results

Overall, the results of the replication part of this study could indeed replicate the findings by Meng et al. (2022), if accounting for the fact that a significantly smaller model with fewer layers was used. A causal effect of individual states could unsurprisingly be observed in the last layers before the target token had to be predicted. The more interesting observation however, as pointed out by Meng et al. (2022) as well, was the effect of patching in clean activations at the early middle layers (layers 5-9) of the model during the processing of the last subject token. The causal tracing analysis revealed that like in Meng et al. (2022), the MLP modules are what is most decisive for the output at those early layers, while the attention modules are more important at the late site. The AIEs for the interesting early middle layers as well as the ATE for the replication and the mathematical application of the CMA can be found in Table 1.

The effects observed during the causal tracing analysis were overall stronger for the mathematical application of the methodology. Generally though, patterns similar to the original applications could be observed when analyzing GPT2-Math. Looking at Table 1 and the heatmaps in Figure 1 displaying the AIE of the hidden states, singled out MLPs, and attention modules, it becomes clear that here as well, not only the clean patching of late states made a considerable impact on the model output. While it is clear to see that the MLP modules in the late sites also seem to store more information on the mathematical operator, the MLP modules at the early to middle layer around layer 5 are what is highly decisive for the model output. Slightly different to the factual association application, for GPT2-Math an effect of the attention modules can be seen at the early layers during the processing of the second number token and not only at the later layers when processing the last token.

Yet, again, the essential role of the MLP modules' computations at the early middle layers is also reflected in the case that they are frozen in their corrupted state when their corresponding state's clean activations are patched into the model. Arguably, this effect is on average a little less pronounced in the math example, where the AIE is about half for freezing the MLP, while in the factual association example the AIE nearly disappears completely. The causal effects of the states with their attention or MLP modules frozen in corrupted states visualized as a bar graph over the layers for both models can be found in the appendix alongside the heatmaps for the replication of Meng et al. (2022) for comparison.

Model	ATE	AIE Hidden	AIE MLP	AIE Attn
GPT2-medium	21.6%	6.4%	6.9%	1.5%
GPT2-Math	60%	51%	51.4%	5.2%

Table 1: AIEs at the early middle layers during processing of the last subject token / mathematical operator token and ATE for both models

The ROME part of the present study showed similar results to the original study for the replication part. For the math application of ROME, novel, wrong mathematical associations like "1 plus 2 equals 4" or "5 plus 2 equals 10" were added into the model at layer 5 and follow-up generation prompts were given to the model to investigate whether the representation of the mathematical operator or one of the numbers would change for other exercises as well. All of the edits were successful, leading the model to now predict the desired wrong target token. Additionally, some effects of generalization to other mathematical exercises could be observed. For example, the prompt "1 plus 3 equals" led to the output "1 plus 3 equals 5" and "1 plus 4 equals" yielded "1 plus 4 equals 6" after the ROME update on "1 plus 2 equals 4", while the pre-edit predictions gave the mathematically correct output. Also, after an edit was made to include "5 plus 2 equals 10" in the model, the previously correct predictions of the prompts "3 plus 2 equals" and "3 plus 3 equals" now yielded the predictions "3 plus 2 equals 6" and "3 plus 3 equals 9".

4 Discussion and Conclusion

First and foremost, the previously presented results can overall be taken to positively answer the overarching research question. Using causal mediation analysis, it was indeed possible to locate knowledge about mathematical operators within a fine-tuned GPT2-Math model and some exemplary rank-one model edits further supported this localization claim. The presented results suggest a localization of knowledge about mathematical operators within the GPT2-Math model at the site of the MLP modules in the early to middle layers, specifically during the processing of the mathematical operator tokens. The causal tracing technique revealed those localizations and the ROME method supported them by showing that model outputs can be successfully edited when only adapting the model weights at that specific location.

The overall very strong effects of the corruptions and re-patching during the second and third run of the CMA that are also reflected in the high ATE and AIEs of the single hidden states for the GPT2-Math, might be explained by how the prompts and training dataset for fine-tuning were structured. Compared to the general factual associations, the math exercises that the GPT2-Math model was trained on and prompted with were a lot more structurally uniform. The number of tokens for the subject of the CMA, i.e. the mathematical operator only varied between 1 and 2 and the rest of the prompts generally followed the same patterns. Additionally, it might be

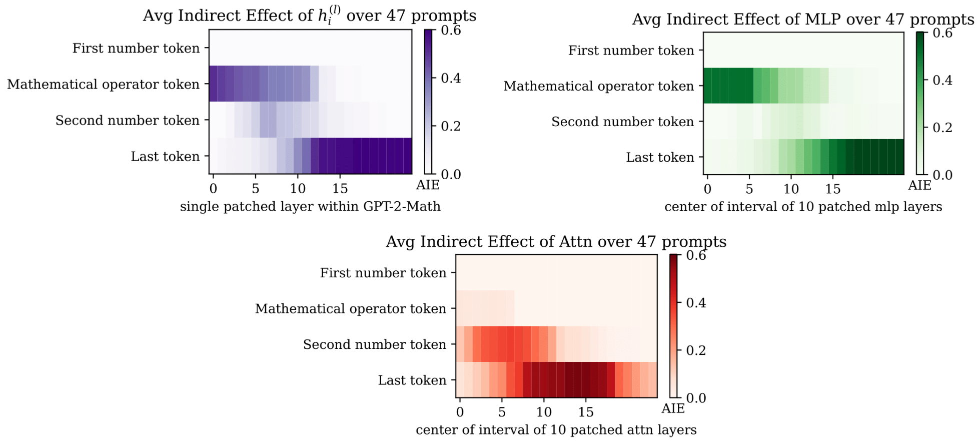

Figure 1: Heatmaps of AIE of whole hidden states, MLP modules, and attention modules in GPT2-Math

argued that mathematical operators are somewhat more universal than subjects of factual associations. While there might be infinitely many subjects, there are only four distinct mathematical operators in this example, a fact that might be reflected in the model representation as well.

It might also be argued that the knowledge about mathematical operators might be slightly less localized than factual associations in the GPT2-medium model. In the case of GPT2-Math the attention modules at the middle layers during the processing of the second number token also seem to play some causal role in recovering the correct output in a corrupted network. However, the effects of those attention modules are substantially lower than the effects of the previously highlighted MLP modules. Additionally, the application of ROME only changed the weights of a single MLP module in layer 5, and was observed to successfully add the mathematical associations into the network.

The results of the ROME applications to single mathematical examples not only supported the found localization of the mathematical operator knowledge in the targeted MLP modules, but also revealed some rather surprising effect of generalization in the interpretation of single numbers and the operators in question. Interestingly, this generalization was not observed when the pre-edit prediction of the model was already producing and incorrect answer. Conversely, for the two cases presented in the results section, it can be hypothesized that the representation of the number 1 was changed more fundamentally by the added faulty math example "1 plus 2 equals 4" demonstrating a change in the relation between numbers. This was not the case for the second number 2 involved in the edit as evidenced by unaffected examples including the number 2 as a second number. In the case of the added faulty example "5 plus 2 equals 10" in which the plus operator now had the effect of a times operator, it was highly interesting to see that a simple small edit could change the interpretation of the operator also for other examples. Importantly, the representation of the times operator did not seem to be affected by this edit.

One limitation of this short study is that the evaluation of the ROME edits was not done more systematically and only using some examples. In the original paper by Meng et al. (2022) a more systematic method is developed to evaluate the generalization and specificity of the edits more thoroughly. However, this evaluation is tied to a specifically developed dataset using several counterfacts and the effects on nearby subjects and paraphrase scores. Developing a similar kind of system for the present case of mathematical examples, while theoretically possible, would exceed the frame of this project by far.

Another limitation of this small study that needs to be kept in mind is the fact that the GPT2-Math model was specifically fine-tuned on the kind of mathematical examples to expect for the causal mediation analysis. This may have led to some fundamental changes in the model's overall abilities. Given that GPT2-medium did not achieve close to acceptable performance on the adapted SVAMP dataset, the fine-tuning was quite aggressive and the resulting model should not be seen as an adaptation of GPT2-medium, but rather as a whole new specialized model for elementary level math tasks. It would therefore be highly interesting to see whether similar results can be found in a model like GPT4 that is already able to solve the prediction tasks well. For the time being, the present study should be viewed as an initial proof of concept, that CMA and ROME methods can indeed be used to interpret representations of mathematical operators.

Summing up, it can be said that the presented methodology holds great promises for future applications on bigger GPT-like models and can be of great help in explaining and interpreting mathematical abilities of LLMs.

Acknowledgements & Data Availability

The methodology used was mainly inspired from Meng et al. (2022). ChatGPT 4o was used to assist with the debugging of python code.
All code and data can be found under `https://github.com/khnhenriette/ADLfinal`.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mariam Bangura, Kristina Barabashova, Anna Karnysheva, Sarah Semczuk, and Yifan Wang. 2023. Automatic generation of german drama texts using fine tuned gpt-2 models. *arXiv preprint arXiv:2301.03119*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2024. Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, 36.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

Mohammad Hosseini, Catherine A Gao, David M Liebovitz, Alexandre M Carvalho, Faraz S Ahmad, Yuan Luo, Ngan MacDonald, Kristi L Holmes, and Abel Kho. 2023. An exploratory survey about using chatgpt in education, healthcare, and research. *Plos one*, 18(10):e0292216.

Jieh-Sheng Lee and Jieh Hsiang. 2020. Patent claim generation by fine-tuning openai gpt-2. *World Patent Information*, 62:101983.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36. ArXiv:2202.05262.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. *arXiv preprint arXiv:2204.05660*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Mengxue Zhang, Zichao Wang, Zhichao Yang, Weiqi Feng, and Andrew Lan. 2023. Interpretable math word problem solution generation via step-by-step planning. *arXiv preprint arXiv:2306.00784*.

Wei Zhang, Chaoqun Wan, Yonggang Zhang, Yiu-ming Cheung, Xinmei Tian, Xu Shen, and Jieping Ye. 2024. Interpreting and improving large language models in arithmetic calculation. *arXiv preprint arXiv:2409.01659*.

Appendix

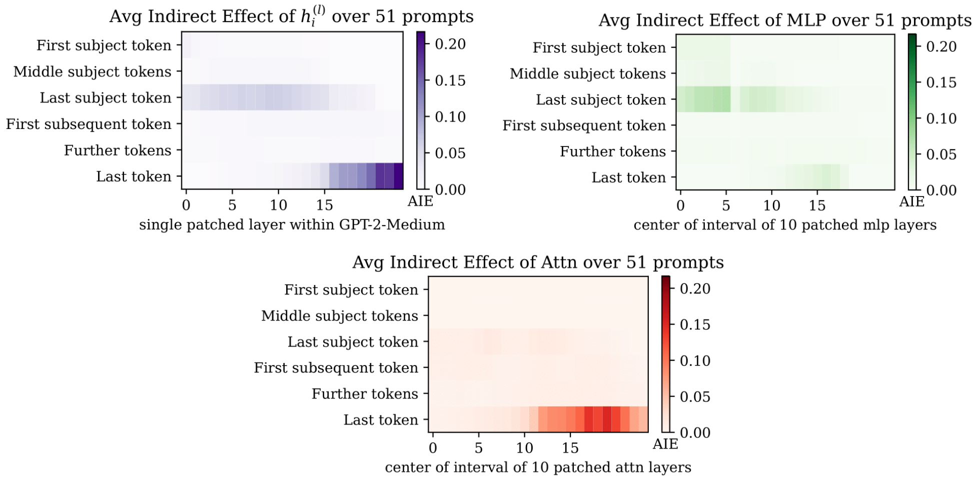

Figure 2: Heatmaps of AIE of whole hidden states, MLP modules, and attention modules for the replication with GPT2-medium

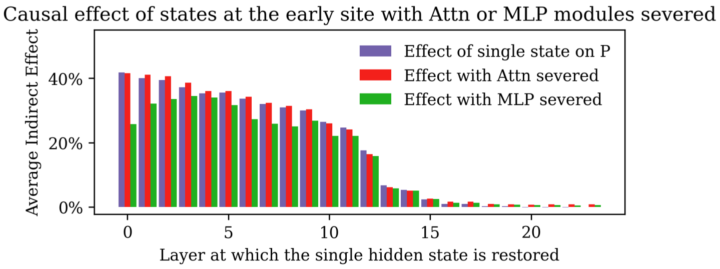

Figure 3: Causal effects for GPT2-Math

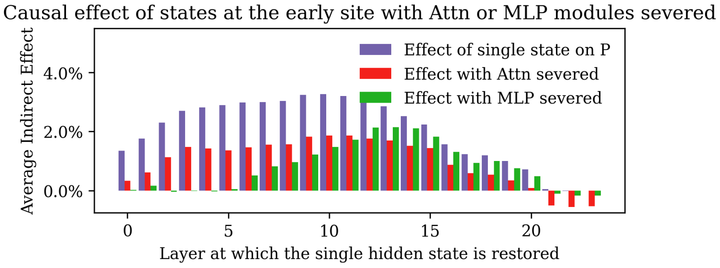

Figure 4: Causal effects for GPT2-medium