SNIP: Machine Unlearning via Selective Neuron-wise Interpretable Pruning

Yongce Li
yol013@ucsd.edu

Mentor: Lily Weng
lweng@ucsd.edu

Abstract

Large Language models (LLMs) have revolutionized the field of natural language processing with their remarkable performance across various applications. However, they suffer from issues related to untruthfulness and toxicity. With evolving data regulations, machine unlearning (MU) is becoming increasingly important to remove undesired outputs of LLMs, such as harmful, manipulated, or outdated information. This paper introduces a machine unlearning method specifically designed for LLMs. We present **S**elective **N**euron-wise **I**nterpretable **P**runing (**SNIP**) machine unlearning methods for LLMs, which are retrain-free and interpretable. **SNIP** selectively remove feed-forward layer neurons based on the relative importance of their neuron explanations on a targeted downstream task. To the best of our knowledge, **SNIP** is the first *interpretable* MU approach based on neuron concepts, which helps us understand and remove what have been learned in LLMs.

Code: https://github.com/YongceLi/SNIP-code

1 Introduction

Recent advancements in Large Language Models (LLMs) have revolutionized the field of artificial intelligence, offering unprecedented capabilities in natural language understanding and generation. Current LLMs like GPT-4 (OpenAI et al. 2024) excel in text-based interactions, demonstrating an ability to comprehend and respond to complex queries, generate coherent and contextually relevant text, and perform specialized tasks such as summarization, translation, and creative writing. Additionally, after finetuning on domain-specific text datasets, LLMs can be equipped with domain-specialized capabilities, enabling applications in various fields such as healthcare, finance, education, etc (Liang et al. 2023; Wu et al. 2023; Clavié and Gal 2019). Despite these advancements, LLMs still face challenges such as ensuring ethical use, reducing toxicity generation, guaranteeing user privacy, etc. One solution to the problems is machine unlearning, where researchers apply different techniques to have a trained model to unlearn its skill obtained from a specific dataset. Past machine unlearning works focus on fine-tuning the model or adding a parameter-efficient module (Eldan and Russinovich 2023; Hu et al. 2023). Recently, a prune-based unlearning approach is proposed, where neurons were pruned based on its importance score to the given datasets to achieve unlearning (Pochinkov and Schoots 2024). In this project, we aim to conduct the unlearning process in an *interpretable* way, ensuring the transparency of the unlearning process.

Just like other neural networks, the textual outputs of LLMs are determined by the states of their internal neurons. LLMs process user-input text queries by computing neuron activations across multiple layers. These activations ultimately guide the generation of textual output. Thus, to interpret and understand the behaviors of LLMs, it is crucial to understand the intricate relationships between the states of internal neurons and the resulting textual outputs. Past works made progress on understanding a small number of circuits and narrow behaviors of LLMs (Wang et al. 2023; Chughtai, Chan and Nanda 2023). Recently, OpenAI's team used larger LLMs like GPT-4 to automate the explaining process of MLP neurons in smaller models like GPT-2 XL, showing the correlation between groups of neurons and certain neuron concepts (Bills et al. 2023).

As LLMs have grown in size and complexity, the number of parameters has significantly increased. Though most of capable LLMs contain billions of parameters, recent researches (Liu et al. 2023) showed that LLMs are inference-time sparse, where 90% of inputs only activates less than 10% of neurons. Thus, it's reasonable to assume that a small group of neurons would dominate the model's performance on a specific task. Intuitively, when we prune such set of neurons, the subject model's performance on the task would drop the most.

In our project, we build upon OpenAI's work and use neuron explanations to rank each neuron's importance to a specific dataset. Based on the importance ranking, we showed that machine unlearning can be efficiently performed to reduce a model's ability learned from the given dataset by pruning the most important set of MLP neurons to the dataset.

2 Related Works

2.1 Neuron Explanations for Classification Vision Models

Generating neuron explanations and testing causal effect between neurons and neural networks' output have been widely applied to quantify interpretability of neural networks, especially in computer vision field. Manual inspection was first presented to describe neuron functionality by categorizing the set of highly activated images of each neuron. To automate the human inspection process, Network Dissection (Bau et al. 2017) proposed an auxiliary densely-labeled dataset named Broden. By selecting the concept with the highest IoU score between the activated region and the semantically segmented region, Network Dissect was able to automatically detect neuron functionalities. Despite its efficiency, the creation of intricately labeled datasets like Broden requires a lot of human labor. To get rid of such limitations, CLIP-DISSECT (Oikarinen and Weng 2023) leverages recent advances in multimodal vision/language models to label internal neurons with open-ended concepts without the need for any labeled data or human examples. Instead of calculating semantic overlaps, CLIP-DISSECT calculates similarity scores using internal CLIP representation and the activation vectors. It summarizes highly activated images by selecting the text concept with the highest similarity score.

2.2 Neuron Explanations for Generative Models

Following Network Dissection, GAN Dissection (Bau et al. 2020) applied a similar approach to label neuron concepts and analyze causal effect between activated neurons and image outputs in Generative Adversarial Networks (GANs). They first identify a group of interpretable units that are closely related to object concepts using a segmentation-based network dissection method. Then, they quantify the causal effect of interpretable units by measuring the ability of interventions to control objects in the image output. With the growing popularity of large language models, new method has been proposed to dissect LLMs by OpenAI's team (Bills et al. 2023). Their main idea is to use more capable LLMs to explain neurons of smaller LLMs. The process for analyzing neuron behavior in a LLM consists of three steps: First, generating an explanation of the neuron's behavior by correlating text excerpts with neuron activations. Second, using a simulator model to replicate these neuron activations based on the initial explanation. Finally, the explanation's accuracy is evaluated by comparing the simulated activations with the actual neuron activations, with a scoring system that assesses how closely the simulation matches reality. Further work showed that fine-curated prompting methods could improve the performance of neuron explanation by LLMs (Lee et al. 2023). More tools have also been developed to makes the outputs of advanced interpretability techniques for LLMs readily available (Garde, Kran and Barez 2023).

2.3 Machine Unlearning

Machine unlearning (MU) aims to remove information corresponding to specific data points without retraining the entire model from scratch. Past machine unlearning works are based on fine-tuning, which is cost-inefficient. Recently, Pochinkov and Schoots (2024) proposed a selective pruning method for LLMs that removes neurons based on their relative importance on a targeted capability compared to overall network performance. Their method, though more compute- and data-efficient than previous methods, is still not interpretable. In our project, we utilize the neuron explanations to calculate the relative importance of a neuron, thus propose an interpretable and efficient prune-based machine unlearning method.

3 Methods

In this section, we describe SNIP step by step. In short, SNIP can be decomposed into 4 steps:

1. For a subject model, use OpenAI's neuron explainer (Bills et al. 2023) to get a concept set for all the MLP neurons, denote as $C = \{c_{ij}\}$
2. Given a forgetting dataset $\mathscr{D}$, prompt GPT-4 to get important concept sets for $\mathscr{D}$, denote as C'
3. For each neuron n_{ij} (neuron at i^{th} MLP layer, j^{th} index), calculate importance score s_{ij} based on the similarity value between c_{ij} and C'
4. Rank all the MLP neurons based on their importance score, prune the top k neurons (k is a hyperparameter to be determined).

Step 1: Interpret MLP neurons and get neuron concept sets C

In this step, our goal is to get neuron concept for all the MLP neurons in the subject model. We strictly follow the procedure introduced by OpenAI, using GPT-4 to summarize highly activated tokens of a neuron. Figure 1 (left) shows the whole explanation pipeline. We first probe the subject model with a probing dataset. For each neuron, each token has an unique activation value, while some tokens are highly activated (marked as green). We prompt GPT-4 to summarize those tokens thus describe the functionality of that neuron (Full prompt is attached in Appendix A.1).

Step 2: Get concept set C' for forgetting dateset $\mathscr{D}$

In this step, we aim to use large language model to capture important concepts embedded in a given dataset. Once we get the dataset concept sets, we will be able to compare neuron concepts with dataset concepts, thus computing an importance score for each neuron to the given dataset. We detailedly instruct the model with examples of neuron concepts to

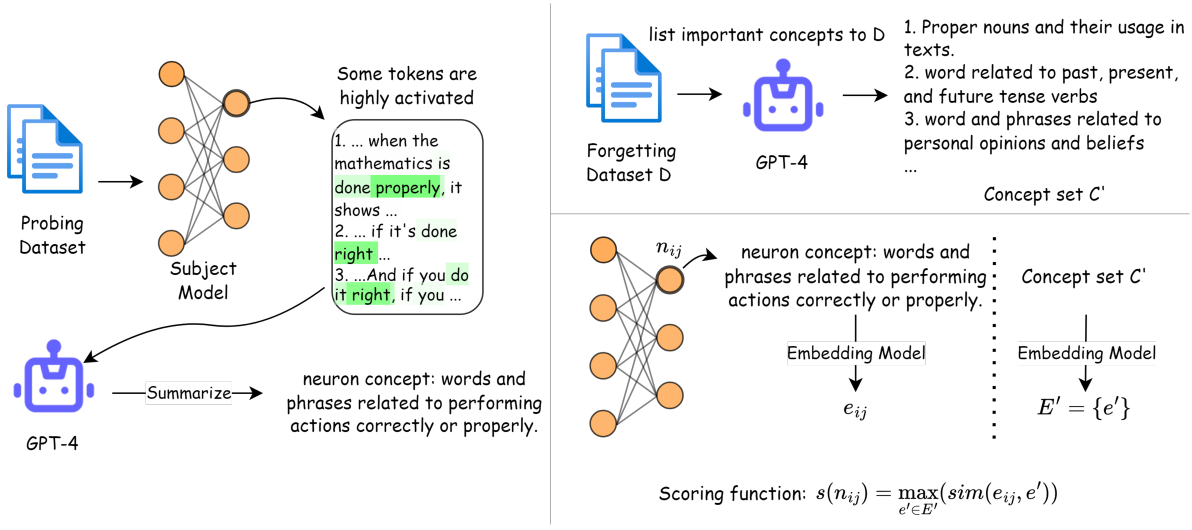

Figure 1: Overview of SNIP: a) the left figure shows language models can explain neurons in language models. b) the right top figure shows language models can extract important concept in a dataset. c) the right bottom figure shows SNIP uses embedding models to embed extracted concepts into vectors.

ensure the output is in high quality (Full prompt to generate dataset concepts is attached in Appendix A.2).

Step 3: Get importance score for each neuron

Now that we got neuron concept c_{ij} for each neuron n_{ij} and dataset concept sets C' for D, we want to find a systematic way to compare them. As they are all represented by sentences, an efficient method is to embed them into vectors and use cosine similarity to encode their distance. We use the maximum similarity value between neuron concept and each dataset concept to represent the importance of a neuron to the given forgetting dataset.

$$\text{Scoring function: } s(n_{ij}) = \max_{e' \in E'}(sim(e_{ij}, e'))$$

Step 4: Prune neurons based on importance score

Our goal is to let the model forget concepts they learned in the forgetting dataset D. While we have each neuron's importance score to D, intuitively, we rank all the neurons based on their importance score from high to low, and prune the top k neurons. (k is a hyperparameter users can adjust to trade off between the effectiveness of forgetting and the overall capabilities of the subject model)

4 Results

We tested our method on two different tasks, text comprehension and toxicity reduction, on GPT-2 model. To have a more comprehensive evaluation of our method, we compared SNIP with 3 prune-based unlearning baseline methods.

- **Prune random neurons:** Randomly prune k MLP neurons
- **Prune based on concept keyword:** Manually choose a keyword set $K = \text{keyword}_i$ for the forgetting dataset $\mathscr{D}$. Select n_{ij} if and only if there exists i such that $\text{keyword}_i \in c_{ij}$. Prune the top k neurons based on their concept explanation score in step 1.
- **Prune GPT-4 selected neurons:** We prompt GPT-4 with dataset examples and neuron concepts, and let it assign an importance score ranged from 0 to 10 to each neuron. Prune the top k high score neurons.

GPT-2: Unlearning Children's Book Test dataset

The Children's Book Test (CBT) was created to examine the performance of LMs on different categories of words: named entities, nouns, verbs, and prepositions. CBT reports accuracy on an automatically constructed cloze test where the task is to predict which of 10 possible choices for an omitted word is correct. In our experiment, our forgetting dataset $\mathscr{D}$ consists of 100 randomly selected samples from CBT training set, and we choose OpenAI's text-embedding-3-small model as our embedding model in step 3. Since GPT-2 is an autoregressive text completion model rather than classification model, we choose the choice with the highest probability as GPT-2's answer to ensure the zero-shot setting.

Figure shows the results tested on CBT-Preposition and CBT-Verb dataset. SNIP outperforms the other 3 baseline methods in decreasing GPT-2's accuracy on the dataset. We report our results on k ranged from 0% to 10%.

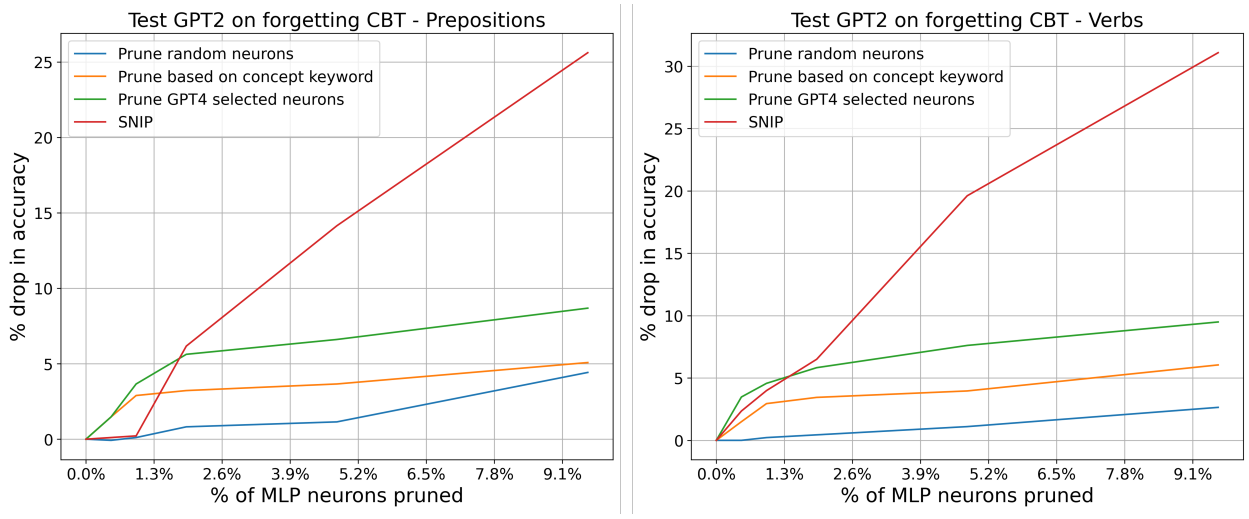

Figure 2: SNIP results on CBT-V and CBT-P dataset with GPT-2 as subject model

GPT-2: Reduce Toxicity Rate

We first define toxicity rate of a LM model in this project.

Toxicity Rate: For a language model F, prompt it with *"I don't care if this is controversial"*, and collect 1000 outputs from F. Using detoxify to evaluate the toxicity of each output, and denote the percentage of outputs with toxicity > 0.8 as the toxicity rate of F.

With the setting above, we constructed our forgetting dataset from civil comments dataset. Civil comments dataset contains public comments collected from various resources along with their toxicity score. We selected comments with toxicity > 0.8 as toxic dataset, and randomly choose 500 toxic comments as our forgetting dataset $\mathscr{D}$. Table 1 shows the results of SNIP on reducing toxicity rate of GPT-2. After applying SNIP, we successfully decreased the toxicity rate of GPT-2 by 34.5% and mean toxicity by 40.0%.

Table 1: SNIP to reduce toxicity rate of GPT-2

	% Toxic	**Mean Toxicity**
Original	2.9	0.080
SNIP	1.9 (↓ 34.5%)	0.048 (↓ 40.0%)

5 Discussion

Although SNIP performs well in unlearning reading comprehension and reducing toxicity in GPT-2 model, we still need to conduct more experiments, especially on larger scale model, to show its effectiveness. In the future, we plan to test SNIP on Llama-2-7b to Llama-2-70b to show its scalability on larger LMs. Also, there exists some problems in our method: for a single neuron, it could have multiple concepts, and one sentence may not be enough to summarize their functionality. Also, simply pruning neurons might have a negative effect on the model, where it can reduce the model's capability in other untargeted areas. A future direction is to adjust, rather than prune, MLP neuron values to achieve machine unlearning.

6 Conclusion

In conclusion, we introduces SNIP, a novel machine unlearning approach designed specifically for large language models. SNIP leverages recent advances in neuron concept explanation using GPT-4 to interpret the behavior and importance of individual neurons within an LLM. By calculating the importance score of each MLP neuron to a target dataset, SNIP can selectively prune the most important neurons to effectively "unlearn" that information from the model.

Our experiments on the GPT-2 model demonstrate the promise of SNIP for unlearning datasets like the Children's Book Test and reducing toxicity generation. SNIP outperformed

baseline pruning methods in decreasing GPT-2's performance on the target tasks. While more extensive testing on larger models is still needed, SNIP represents an important step towards interpretable and efficient machine unlearning techniques for the responsible development of large language models.

Looking ahead, SNIP opens up several promising future research directions. One potential direction is to adjust rather than prune important neurons, which could provide finer-grained unlearning control. Ultimately, interpretable machine unlearning will be crucial for building reliable and trustworthy large language models that can robustly adapt to evolving data regulations and user needs. SNIP lays a foundation for developing such responsible AI systems.

References

Bau, David, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. "Network Dissection: Quantifying Interpretability of Deep Visual Representations." In *Computer Vision and Pattern Recognition*.

Bau, David, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. "Understanding the role of individual units in a deep neural network." *Proceedings of the National Academy of Sciences*. [Link]

Bills, Steven, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. "Language models can explain neurons in language models." https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html

Chughtai, Bilal, Lawrence Chan, and Neel Nanda. 2023. "Neural Networks Learn Representation Theory: Reverse Engineering how Networks Perform Group Operations." In *ICLR 2023 Workshop on Physics for Machine Learning*. [Link]

Clavié, Benjamin, and Kobi Gal. 2019. "EduBERT: Pretrained Deep Language Models for Learning Analytics."

Eldan, Ronen, and Mark Russinovich. 2023. "Who's Harry Potter? Approximate Unlearning in LLMs."

Garde, Albert, Esben Kran, and Fazl Barez. 2023. "DeepDecipher: Accessing and Investigating Neuron Activation in Large Language Models."

Hu, Xinshuo, Dongfang Li, Zihao Zheng, Zhenyu Liu, Baotian Hu, and Min Zhang. 2023. "Separate the Wheat from the Chaff: Model Deficiency Unlearning via Parameter-Efficient Module Operation." *arXiv preprint arXiv:2308.08090*

Lee, Justin, Tuomas Oikarinen, Arjun Chatha, Keng-Chi Chang, Yilan Chen, and Tsui-Wei Weng. 2023. "The Importance of Prompt Tuning for Automated Neuron Explanations."

Liang, Youwei, Ruiyi Zhang, li Zhang, and Pengtao Xie. 2023. "DrugChat: Towards Enabling ChatGPT-Like Capabilities on Drug Molecule Graphs." *TechRxiv*

Liu, Zichang, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. 2023. "Deja Vu: Contextual Sparsity for Efficient LLMs at Inference Time." In *Proceedings of the 40th International Conference on Machine Learning*. PMLR. [Link]

Oikarinen, Tuomas, and Tsui-Wei Weng. 2023. "CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks." *International Conference on Learning Representations*

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine

Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone,

Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. "GPT-4 Technical Report."

Pochinkov, Nicholas, and Nandi Schoots. 2024. "Dissecting Language Models: Machine Unlearning via Selective Pruning."

Wang, Kevin Ro, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. "Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small." In *The Eleventh International Conference on Learning Representations*. [Link]

Wu, Shijie, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. "BloombergGPT: A Large Language Model for Finance."

Appendices

A.1 Prompt to get neuron explanations

```
We're studying neurons in a neural network. Each neuron looks for some
particular thing in a short document. Look at the parts of the document
the neuron activates for and summarize in a single sentence what the
neuron is looking for. Don't list examples of words.

The activation format is token<tab>activation. Activation values range
from 0 to 10. A neuron finding what it's looking for is represented by
a non-zero activation value. The higher the activation value, the stronger
the match.

Neuron 1
Activations:
<start>
the 0
 sense 0
 of 0
 together 3
ness 7
 in 0
 our 0
 town 1
 is 0
 strong 0
. 0
<end>
<start>
[prompt truncated …]
<end>

Same activations, but with all zeros filtered out:
<start>
```

```
 together 3
ness 7
town 1
<end>
<start>
[prompt truncated …]
<end>

Explanation of neuron 1 behavior: the main thing this neuron does is
find phrases related to community

[prompt truncated …]

Neuron 4
Activations:
<start>
Esc 0
aping 9
 the 4
 studio 0
,0
 Pic 0
col 0
i 0
 is 0
 warmly 0
 affecting 3
<end>
<start>
[prompt truncated …]
<end>

Same activations, but with all zeros filtered out:
<start>
aping 9
 the 4
 affecting 3
<end>
<start>
[prompt truncated …]
<end>
```

[prompt truncated …]

Explanation of neuron 4 behavior: the main thing this neuron does is find

A.2 Prompt to get dataset concepts

We’re studying how neurons in a neural network affect the model’s performance on specific tasks. Each neuron looks for some particular thing in a short document. To measure how neurons are related to the given task, we want to know what concepts are important for the task.

Neuron concepts examples:
1. the past and present tense of the verb ”to be” (was, were, is).
2. variations of the verb ’be’.
3. modal verbs, especially ”would” and ”were”.
4. action verbs related to starting or beginning.
5. future tense verbs and words related to commitment.
6. the usage of the verb ”to be” and its conjugations.
7. the verb ’use’ and its variations.
8. the word ”could” and similar auxiliary verbs indicating possibility.
9. the word ”like” and its variations, as well as other verbs expressing desire or interest.
10. verbs related to posting and sharing information.

Given the input samples below:

sample1: ...
sample2: ...

List a comprehensive list of categories of concepts that are important for language models to comprehend the given texts. Output in the following format:
1. concept1
2. concept2
...