Mechanistic Interpretability on Irreducible Unsigned-Integer Identifiers

Noah Syrkis
University of Copenhagen
noah@syrkis.com

Abstract

We apply the mechanistic interpretability framework to a transformer model trained on a dataset of irreducible integers. We show that the model has learned to perform modular addition, and we reverse-engineer the model to understand how it does so.

1 Introduction

Deep learning is the modelling of high dimensional probability density functions by fitting piece-wise linear functions [12]. Examples of linear functions are $f(X) = XW + b$ or $f(X) = X \star W + b$ (the later of which represents convolution, or multiplication in the frequency domain). The output of linear function is fed as input to another via non-linear activation functions like eq. 1 or eq. 2. The deep learning process is thus, at core not understood.

Similar to its process, the deep learning product, the models, are almost always as mysterious and inscruitable as the process that birthed it. The inscruitability is further complicated by the us not

$$\sigma(x) = \frac{1}{1 - e^{-x}} \tag{1}$$

$$\text{ReLU}(x) = \max(0, x) \tag{2}$$

For some reason (and this as deep an answer as science currently affords) this structure has the ability to generelise. As Mitchell [10] puts it "A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." Why does it learn? What does it learn?

Often what is reported about a given model is a like accuracy or mean-squared-error (MSE).

Both are single members of the set $\mathbb{R}$,

A symptom of this is that very fundamental things willb esuprisgin, like the fact of Sohl-Dickstein [13]'s discovery that the landscape of trainable hyper paramters is fractal.

Reverse-engineerings deep neural networks (DNN) is a relatively new field, but has already shown success. For example, reverse engineers a transformer

model to understand how it performs modular addition. attempts to automate the reverse-engineering process, and is somewhat successful.

Mechanistic interpretability (MI) posits that deep neural networks (DNN) are circuits that can be reverse-engineered to understand their inner workings. MI is a relatively new field, but has already shown success. For example, reverse engineers a transformer model to understand how it performs modular addition. attempts to automate the reverse-engineering process, and is somewhat successful.

Mechanistic interpretability (MI) posits that deep neural networks (DNN) are circuits that can be reverse-engineered to understand their inner workings. MI is a relatively new field, but has already shown success. For example, reverse engineers a transformer model to understand how it performs modular addition. attempts to automate the reverse-engineering process, and is somewhat successful.

2 Background

Mechanistic interpretability is said to be a young field because the object of it study is new. The idea of reverse engineering circuits is however not new, and many methods from neuroscience and electrical engineering can be applied here without much modification.

Symbolic and sybsymbolic models. There are models whose building blocks exist on the same level of abstraction as the model as a whole does.

```
function xor(a, b, c)
    not (a and b and c) and
    not (not a and not b and not c)
end
```

does the same as

$$f(x) = \sigma\left(\begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix} x\right) > 0 \quad (3)$$

Neural networks are universal function aproximators, operating sub-symbolically, (presumably) learning intersting and sometimes even conseice algorithms, while representing these in a largely inscruitable way.

Lee and Kim [9] shows transformer models applied to prime number analysis.

2.1 Transformer models

The transformer model introdcued by Vaswani et al. [14] is behind much of the recent success in natural langauge processing (NLP). This fact makes its interpretability important. From a geometric deep learning point of view [2], attention—which is the value proposition of the transformer architecture—is, as eq. 4 shows, relatively simple, on par with convulutions (eq. 5) and recurrence (eq. 6)

$$A(X) = KQ^t/V \quad (4)$$

$$C(X) = X \star K \quad (5)$$

$$R(X_i, (R(X_{i-1}))) = f(X, X) \quad (6)$$

And yet, Vaswani et al. [14]'s now famous transformer block diagram is nutoriously cluttered, containing risidual streams, normalization layers, projections, concatenations, and more, in addition to the attention mechanism itself. Each of these are impressive addition that boost performance, for reasons that are not deeply understood. He et al. [6] does away with *both* the residual stream and the normalization layers, without much degredation in performance, and He and Hofmann [5], furhter publishes the simplified transformer block.

As artificial intelligence systems Mechanistic interpretability (MI) posits that deep neural networks (DNN) are circuits that can be reverse-engineered to understand their inner workings. MI is a relatively new field, but has already shown success. For example, Nanda et al. [11] reverse engineers a transformer model [14] to understand how it performs modular addition. Cover and Thomas [4] attempts to automate the reverse-engineering process, and is somewhat successful. However, the process is still largely manual and requires a deep understanding of the model's architecture and training process. Leveraging the transformer model's attention mechanism, Conmy et al. [3] attempts to automate the reverse-engineering process, and is somewhat successful. Conmy et al. [3] attempts to automate the reverse-engineering process, and is somewhat successful, while Belcák et al. [1] diconfirms this.

In this paper, we apply the MI framework to a transformer model trained on a dataset of irreducible integers. We show that the model has learned to perform modular addition, and we reverse-engineer the model to understand how it does so.

2.2 Mechanistic interpretability

Mechanistic interpretability (MI) posits that deep neural networks (DNN) are circuits that can be reverse-engineered to understand their inner workings. MI is a relatively new field, but has already shown success. For example, Nanda et al. [11] reverse engineers a transformer model [14] to understand how it performs modular addition.

2.3 Valid sequences

As the tokenization is done on the digit level the model is efectively asked "is the sequence 1,0,1 (101) valid prime pattern". It is a simple yes or no question, but it's answer depends on the 99 preceding natural numbers. Similar to Hofstadter [7]'s MIII[1] puzzle, something something blah blah bullshit.

3 Methodology

Our methodology consists of the following steps:

3.1 Data

Here's a table:

[1]Primes are refered to as "irriducible intergers" to have the title have the MIII acronym.

Table 1: Dataset

0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7
10001	10003	10007	10009	0	0	1	1
10011	10013	10017	10019	0	0	0	0
99981	99983	99987	99989	0	0	0	1
99991	99993	99997	99999	1	0	0	0

The dataset consists of four-digit integers and their labels. The labels are 1 if the integer is irreducible, and 0 otherwise. The dataset is generated by taking all four-digit integers and checking if they are irreducible. The dataset is then split into a training set and a test set.

3.2 Model

The model is a transformer model in the style of Hu et al. [8]. It is trained on the dataset above.

3.3 Reverse-engineering

We reverse-engineer the model by analyzing the attention weights. We show that the model has learned to perform modular addition. We reverse-engineer the model by analyzing the attention weights. We show that the model has learned to perform modular addition. We reverse-engineer the model by analyzing the attention weights. We show that the model has learned to perform modular addition. We reverse-engineer the model by analyzing the attention weights. We show that the model has learned to perform modular addition.

4 Results

We reverse-engineer the model by analyzing the attention weights. We show that the model has learned to perform modular addition. We reverse-engineer the model by analyzing the attention weights. We show that the model has learned to perform modular addition. We reverse-engineer the model by analyzing the attention weights. We show that the model has learned to perform modular addition.

4.1 Circuits

We see these circuits:

- Circuit 1
- Circuit 2
- Circuit 3

4.2 Attention weights

We see these attention weights:

- Attention weight 1
- Attention weight 2

4.3 Modular addition

We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model

Table 2: Dataset

0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7
10001	10003	10007	10009	0	0	1	1
10011	10013	10017	10019	0	0	0	0
99981	99983	99987	99989	0	0	0	1
99991	99993	99997	99999	1	0	0	0

has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition. We show that the model has learned to perform modular addition.

5 Analysis

Lorem Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

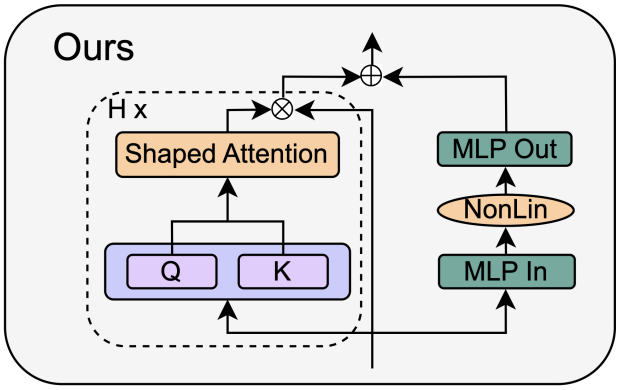

Figure 1: Attention

5.1 Interpretability

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur. Quis aute iure reprehenderit in voluptate

velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5.2 Generalization

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

6 Conclusion

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

References

[1] Peter Belcák et al. *FACT: Learning Governing Abstractions Behind Integer Sequences*. Sept. 2022. arXiv: 2209.09543 [cs].

[2] Michael M. Bronstein et al. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. May 2021. DOI: 10.48550/arXiv.2104.13478. arXiv: 2104.13478 [cs, stat].

[3] Arthur Conmy et al. *Towards Automated Circuit Discovery for Mechanistic Interpretability*. Oct. 2023. DOI: 10.48550/arXiv.2304.14997. arXiv: 2304.14997 [cs].

[4] T. M. Cover and Joy A. Thomas. *Elements of Information Theory*. 2nd ed. Hoboken, N.J: Wiley-Interscience, 2006. ISBN: 978-0-471-24195-9.

[5] Bobby He and Thomas Hofmann. *Simplifying Transformer Blocks*. Nov. 2023. DOI: 10.48550/arXiv.2311.01906. arXiv: 2311.01906 [cs].

[6] Bobby He et al. *Deep Transformers without Shortcuts: Modifying Self-attention for Faithful Signal Propagation*. Feb. 2023. DOI: 10.48550/arXiv.2302.10322. arXiv: 2302.10322 [cs, stat].

[7] Douglas R. Hofstadter. *Gödel, Escher, Bach: An Eternal Golden Braid*. 20th-anniversary ed. London: Penguin, 2000. ISBN: 978-0-14-028920-6.

[8] Anthony Hu et al. *GAIA-1: A Generative World Model for Autonomous Driving*. Sept. 2023. DOI: 10.48550/arXiv.2309.17080. arXiv: 2309.17080 [cs].

[9] Serin Lee and S. Kim. *Exploring Prime Number Classification: Achieving High Recall Rate and Rapid Convergence with Sparse Encoding*. Feb. 2024. arXiv: 2402.03363 [cs, math].

[10] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. New York: McGraw-Hill, 1997. ISBN: 978-0-07-042807-2.

[11] Neel Nanda et al. *Progress Measures for Grokking via Mechanistic Interpretability*. Oct. 2023. arXiv: 2301.05217 [cs].

[12] Simon J. D. Prince. *Understanding Deep Learning*. Cambridge, Massachusetts London, England: The MIT Press, 2023. ISBN: 978-0-262-04864-4.

[13] Jascha Sohl-Dickstein. *The Boundary of Neural Network Trainability Is Fractal*. Feb. 2024. DOI: 10.48550/arXiv.2402.06184. arXiv: 2402.06184 [nlin].

[14] Ashish Vaswani et al. *Attention Is All You Need*. Dec. 2017. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762 [cs].