One is 1: Analyzing Activations of Numerical Words vs Digits[1]

Mikhail L
Independent Researcher

Neel Nanda, Esben Kran, Fazl Barez

Abstract

Extensive research in mechanistic interpretability has showcased the effectiveness of a multitude of techniques for uncovering intriguing circuit patterns. We utilize these techniques to compare similarities and differences among analogous numerical sequences, such as the digits "1, 2, 3, 4", the words "one, two, three, four", and the months "January, February, March, April". Our findings demonstrate preliminary evidence suggesting that these semantically related sequences share common activation patterns in GPT-2 Small. Experiments and code are available at: https://github.com/wlg100/numseqcont_circuit_expms

Keywords:Mechanistic interpretability, AI safety

1. Introduction

There are many unanswered questions in the field of mechanistic interpretability. In this report, we tackle the problem of locating circuits for "Problem 2.2. Continuing sequences that are common in natural language (E.g, Input: "1 2 3 4" -> Output: "5", days of the week, etc.)" from the following list of open problems (Nanda, 2023a):

200 Concrete Problems In Interpretability Spreadsheet

Our approach begins by applying activation patching and attention pattern analysis to GPT-2 Small outputs of analogous sequential inputs, such as the digits "1, 2, 3, 4", the numerical words "one, two, three, four", and the months "January, February, March, April". Next, we employ methods to deduce attention head copying and writing directions, and to examine how Multilayer Perceptron (MLP) neurons react to both numerical features and to features of sequences mapped onto numerical sequences. From these results, we conjecture a very simple candidate circuit for the digits sequence.

Finally, we perform mean ablation on attention heads that are not part of this candidate circuit. As our exploratory analysis reveals that several analogous sequences share the same important heads and attention patterns, we run mean ablation for these analogous sequences on this one circuit of the digits sequence, and observe how their results differ.

[1] Research conducted at the Apart Research Alignment Jam #10 (Interpretability 3.0), 2023 (see https://alignmentjam.com/jam/interpretability)

Finding a shared pattern (within models) between these analogous sequences is of interest to us because informally speaking, we ask: do there exist "primordial" archetypal circuits that compose with additional components to construct more specific circuits? Is there a way to measure the optimality of representations for more general circuits in relation to the sparsity of specific circuits they interact with? Does this affect how they are arranged as directions or regions in latent space? These questions are beyond the scope of this study, but it acts as a start of an investigation.

2. Test Prompts

We start by testing how GPT-2 Small reacts to sequences of increasing order, which include "canonical" mappings to the natural numbers that are widely accepted by society. These include days of the week, differentiating additive sequences like "2 4 6 8" from multiplicative sequences like "2 4 8 16", and more. We also test on other types of sequences, such as on repeating digits or words. While GPT-2 Small is able to correctly predict the output for some consecutive sequences given only 2 or 3 elements, it is able to barely succeed for more complex sequences such as "2 4 6…" only when given 6 or more elements, and fails on others despite being given many elements to guess an in-context pattern from. Afterwards, we choose these 5 sequences to compare and analyze further: digits, numerical words, months, days of the week, and the alphabet.

3. DLA, Activation Patching, and Attention Patterns

We create a template by slightly re-organizing the Exploratory Analysis Demo notebook (Nanda, 2023b), and run this template on various inputs. This template consists of Direct Logit Attribution (DLA), Activation Patching, and Attention Patterns, which are techniques applied by (Wang et al. 2022) that were initiated by (Meng et al., 2022).

To mitigate the cases where the circuit was specific only to "1 2 3 4" and not to other length-4 digit sequences, we run a dataset of multiple prompts consisting of overlapping sequences such as "2 3 4 5", ranging up to 20. For number words, we use one to ten, as there are numerical words higher than ten that consist of multiple tokens. For letters, we only use 7 prompts, as the model does not complete many letter sequences correctly. These "multi-prompt" results are largely similar to the single-prompt results. The following results are from the notebooks in the repo's folder "actv_patch_small":

Direct Logit Attribution

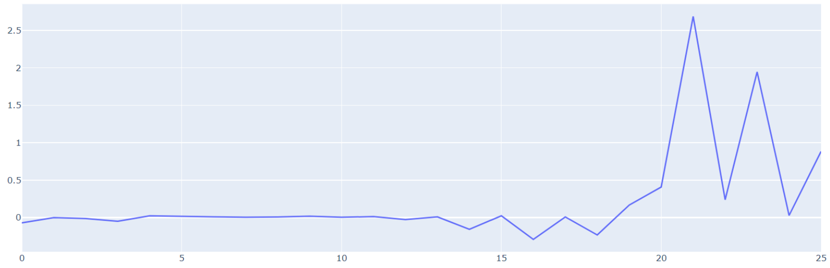


Figure 1 – Logit Difference from each Layer (by attn and MLP) for Digit Prompts

Each analogous sequence had similar DLA results. For instance, each had a similar trend of rising MLP_9 and MLP_10, and falling attn_10 and attn_11, as shown in Figure 1.

Activation Patching

To corrupt the inputs such that the corruption produces (incorrect token logit) > (correct token logit), we consider several possible candidates, including: repeating the last K elements (eg. 1 2 3 3), switching the last and second last elements, and corrupting at the (n-k) position with a repeat or non-number word. We choose to make the last element the same as the 2nd last (eg. 1 2 3 3).

When patching by layer, we observe that each analogous sequence has similar results, with very high values at L9 for both attention and MLP layers. We note that MLP has a higher max range of 0.6 than attention's 0.2, indicating MLP's importance.

Important Attention Heads

Four of the prompt types share many top attention heads with positive restoration found by activation patching. These top heads are shown in Table 1, where the value following the head is the normalized patched logit difference, and the last row shows which heads of that column's top heads differ from the digit's top heads. We denote heads using the labeling structure of Layer.Head (eg. L9H1 := 9.1). Only the alphabet sequence-type differs greatly from the rest, albeit it shares 9.1 as a highly important head. Additionally, we observe that the OV component of head 9.1 is much more important than its QK component.

Digits	NumWords	Months	DaysWeek	Alphabet
9.1: 0.21 7.10: 0.06 10.7: 0.04 8.8: 0.04 0.1: 0.03 8.11: 0.02 6.1: 0.02 0.5: 0.01 9.9: 0.01 11.10: 0.01	9.1: 0.30 0.1: 0.05 8.8: 0.05 7.10: 0.04 6.1: 0.04 0.5: 0.03 8.11: 0.03 10.7: 0.02 11.10: 0.01 0.3: 0.01	9.1: 0.24 7.10: 0.05 0.1: 0.05 8.11: 0.04 6.1: 0.03 10.7: 0.03 0.3: 0.02 0.5: 0.02 8.8: 0.02 5.1: 0.01	9.1: 0.29 8.11: 0.15 0.1: 0.13 7.10: 0.09 6.1: 0.06 8.8: 0.05 10.7: 0.04 6.9: 0.03 0.5: 0.03 5.1: 0.02	10.7: 0.73 9.1: 0.72 11.10: 0.47 11.0: 0.10 9.5: 0.10 6.1: 0.08 8.10: 0.07 5.8: 0.05 0.9: 0.04 0.10: 0.04
N/A	0.3	0.3, 5.1	5.1, 6.9	6 heads

Table A1 - Top positive attention heads and normalized patched logit difference for each input type. Last row: heads of that column's top heads that differ from the digit's top heads

Attention Patterns

The attention patterns of the digits sequences are mostly the same for all its top heads; they attend stronger to more recent numbers. This is unsurprising, as our test prompts find that "corrupting" tokens before the last or second last still allows the model to complete the sequence correctly, indicating those two tokens are by far the most important , though the ones before do still contribute to the logit of the correct output.

For numwords, we find heads such as 6.1, which attend from four to two as seen in Figure 2(a), and also three to two as seen in Figure 2(b). 6.1 also attends to more than just the previous token in months and days, as shown in Figure 2(c). We did not compare with the alphabet sequences as it does not share many heads with the others. (Nanda, 2023c) does not indicate 6.1 is an induction head.

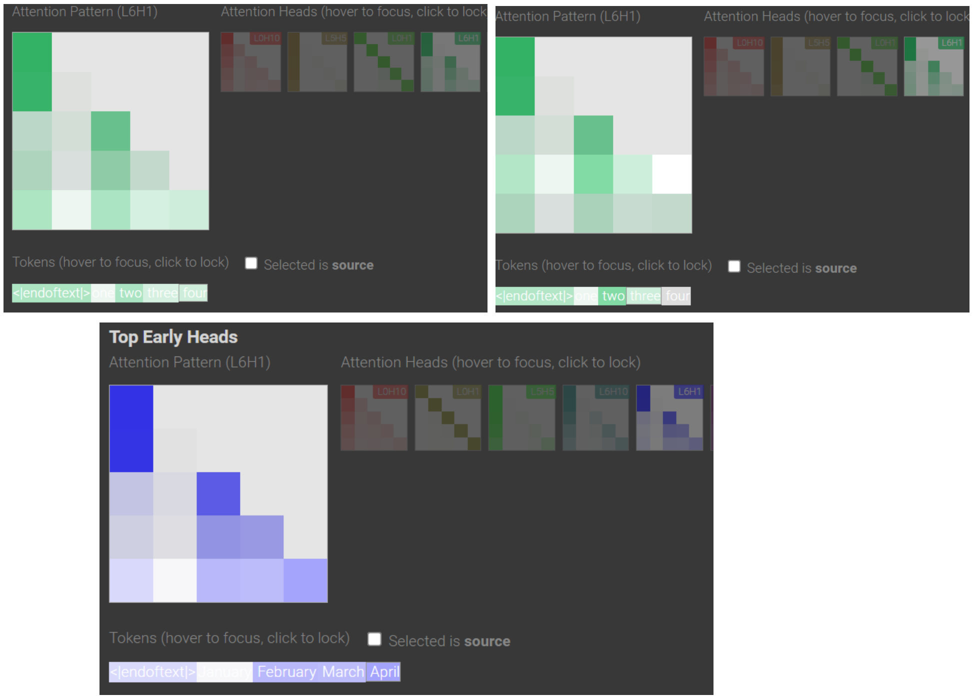

Attention Pattern (L6H1)
Attention Heads (hover to focus, click to lock
L0H10
L3H5
L5H1
L6H1
Tokens (hover to focus, click to lock)
□ Selected is source
<|endoftext|> one two three four
Attention Pattern (L6H1)
Attention Heads (hover to focus, click to lock
L0H10
L3H5
L5H1
L6H1
Tokens (hover to focus, click to lock)
□ Selected is source
<|endoftext|> one two three four
Top Early Heads
Attention Pattern (L6H1)
Attention Heads (hover to focus, click to lock)
L0H10
L0H1
L5H1
L6H10
L6H1
Tokens (hover to focus, click to lock)
□ Selected is source
<|endoftext|> January February March April

Figure 2 – (a) (b) [top] 6.1 for numwords, (c) [bottom] 6.1 for months

Overall, these heads appear to either attend to the previous (n-1) token or the n-2 token. As there are no other 'non-number' words to compare to, it is hard to tell what 'type' of token they are attending to. In Section 6 (Numerical Sequences Among Other Words), we devise an approach to measure what types of tokens the heads attend to.

Comparison to GPT-2 Medium

As GPT-2 Medium is not our main focus, we do not go into depth about it. When we run it on the sequence "2 4 6 8 10 12", we notice that head 11.4 attends to "6", similar to 6.1 in the examples above. More importantly, it has a head 19.1 that is very important, appearing very similar to 9.1. It may be the case that 9.1 from Small and 19.1 from Medium perform similar functions, but we do not investigate further.

4. Attention Heads Analysis: Writing + Copying Directions

These methods re-organize code from (Wang et al., 2022). In Figure 3, the x-axis represents the (<end>, S) value of the QK matrix; that is, the attention that <end> pays to token S. The y-axis represents how much token S contributes to the output value of the attention head (in Figure X, this is 9.1). This measures the strength of the correlation between when <end> attends to S, and how strong the output will be "composed of" S (in the case of copy scores, this means outputting S with high logits).

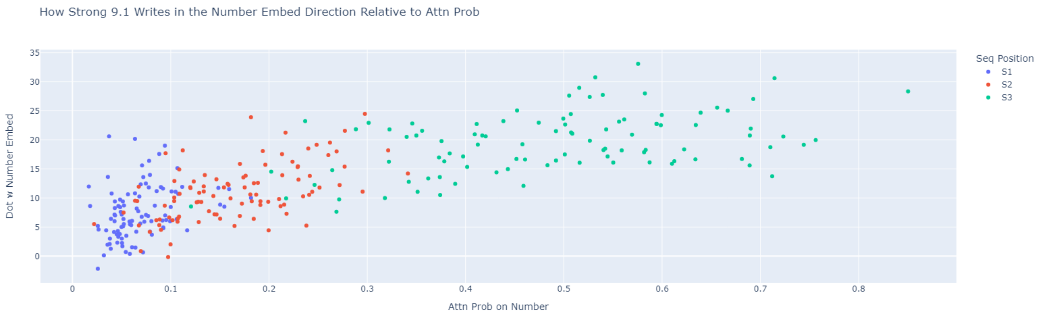
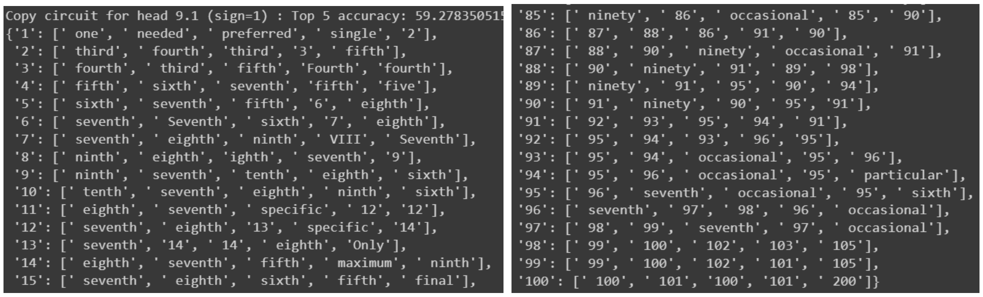


Figure 3 – Writing Direction Scatterplot for tokens in digit sequences, head 9.1 (Correlation: 0.8128, p-value: 8.9942e-70)

We observe there to be strong correlation, indicating that this head 9.1 is detecting numbers.

To check that 9.1 is outputting something to do with numbers, we studied what values are written via the heads' OV matrix. The calculation of this "copy score" is described in (Wang et al., 2022).



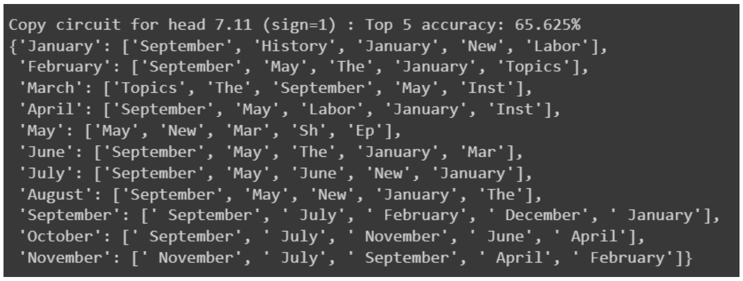
```
Copy circuit for head 9.1 (sign=1) : Top 5 accuracy: 59.27835051
{'1': [' one', ' needed', ' preferred', ' single', '2'],
 '2': [' third', ' fourth', 'third', '3', ' fifth'],
 '3': [' fourth', ' third', ' fifth', 'Fourth', 'fourth'],
 '4': [' fifth', ' sixth', ' seventh', 'fifth', 'five'],
 '5': [' sixth', ' seventh', ' fifth', '6', ' eighth'],
 '6': [' seventh', ' Seventh', ' sixth', '7', ' eighth'],
 '7': [' seventh', ' eighth', ' ninth', ' VIII', ' Seventh'],
 '8': [' ninth', ' eighth', 'ighth', ' seventh', '9'],
 '9': [' ninth', ' seventh', ' tenth', ' eighth', ' sixth'],
 '10': [' tenth', ' seventh', ' eighth', ' ninth', ' sixth'],
 '11': [' eighth', ' seventh', ' specific', ' 12', '12'],
 '12': [' seventh', ' eighth', '13', ' specific', '14'],
 '13': [' seventh', '14', ' 14', ' eighth', 'Only'],
 '14': [' eighth', ' seventh', ' fifth', ' maximum', ' ninth'],
 '15': [' seventh', ' eighth', ' sixth', ' fifth', ' final'],
```

```
'85': [' ninety', ' 86', ' occasional', ' 85', ' 90'],
'86': [' 87', ' 88', ' 86', ' 91', ' 90'],
'87': [' 88', ' 90', ' ninety', ' occasional', ' 91'],
'88': [' 90', ' ninety', ' 91', ' 89', ' 98'],
'89': [' ninety', ' 91', ' 95', ' 90', ' 94'],
'90': [' 91', ' ninety', ' 90', ' 95', '91'],
'91': [' 92', ' 93', ' 95', ' 94', ' 91'],
'92': [' 95', ' 94', ' 93', ' 96', '95'],
'93': [' 95', ' 94', ' occasional', '95', ' 96'],
'94': [' 95', ' 96', ' occasional', '95', ' particular'],
'95': [' 96', ' seventh', ' occasional', ' 95', ' sixth'],
'96': [' seventh', ' 97', ' 98', ' 96', ' occasional'],
'97': [' 98', ' 99', ' seventh', ' 97', ' occasional'],
'98': [' 99', ' 100', ' 102', ' 103', ' 105'],
'99': [' 99', ' 100', ' 102', ' 101', ' 105'],
'100': [' 100', ' 101', '100', '101', ' 200']}
```

Figure 4 – The top-5 tokens output and copy score for digits, head 9.1

Figure 4 shows 9.1 often appears to be outputting a token synonymous to I+1 or higher from a token input I (a number), while sometimes outputting ones before I. Thus, it may function not just as a "copy" head, but as a "next" head. Alternatively but similarly, it may appear to be performing next-token prediction. As there are infinite natural numbers, it may not be memorizing what comes after each number, perhaps after a certain point.

Similar patterns can be seen when passing in month inputs, as seen in Figures 5 and 6.



```
Copy circuit for head 7.11 (sign=1) : Top 5 accuracy: 65.625%
{'January': ['September', 'History', 'January', 'New', 'Labor'],
 'February': ['September', 'May', 'The', 'January', 'Topics'],
 'March': ['Topics', 'The', 'September', 'May', 'Inst'],
 'April': ['September', 'May', 'Labor', 'January', 'Inst'],
 'May': ['May', 'New', 'Mar', 'Sh', 'Ep'],
 'June': ['September', 'May', 'The', 'January', 'Mar'],
 'July': ['September', 'May', 'June', 'New', 'January'],
 'August': ['September', 'May', 'New', 'January', 'The'],
 'September': [' September', ' July', ' February', ' December', ' January'],
 'October': [' September', ' July', ' November', ' June', ' April'],
 'November': [' November', ' July', ' September', ' April', ' February']}
```

Figure 5 - Month inputs on 7.11, Copy Score and top-5 tokens

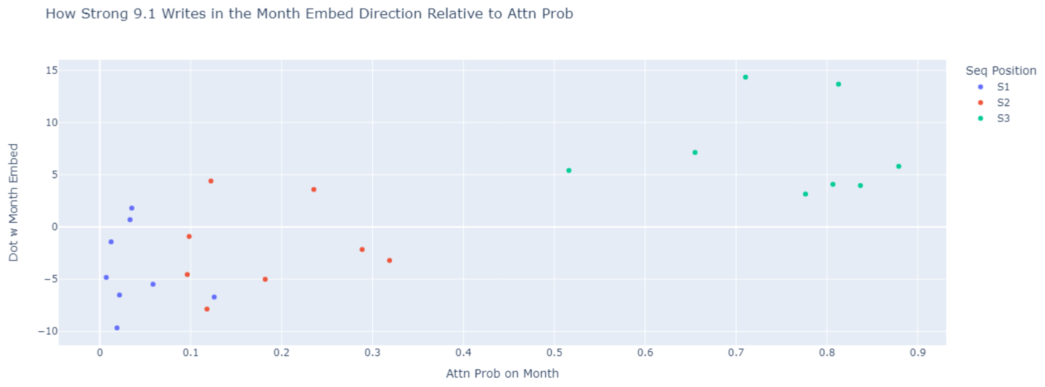


Figure 6 – Writing Direction Scatterplot for tokens in month sequences, head 9.1 (Correlation: 0.9307, p-value: 4.3906e-11)

We plan to modify this code to measure "next" score in addition to copy score in the future, which will provide a more accurate picture of 9.1's importance, and then apply these methods to more heads and analogous sequences to further decipher this circuit's functionalities.

5. MLP Analysis

This analysis uses code re-organized from (Miller et al., 2023). We utilize activation patching on individual neurons to find important neurons, as displayed in Figure 7. Using multi-prompts for digits, we look at L9 and L10 to find two that stand out as "number neurons":

- Layer 9: 934
- Layer 10: 1721

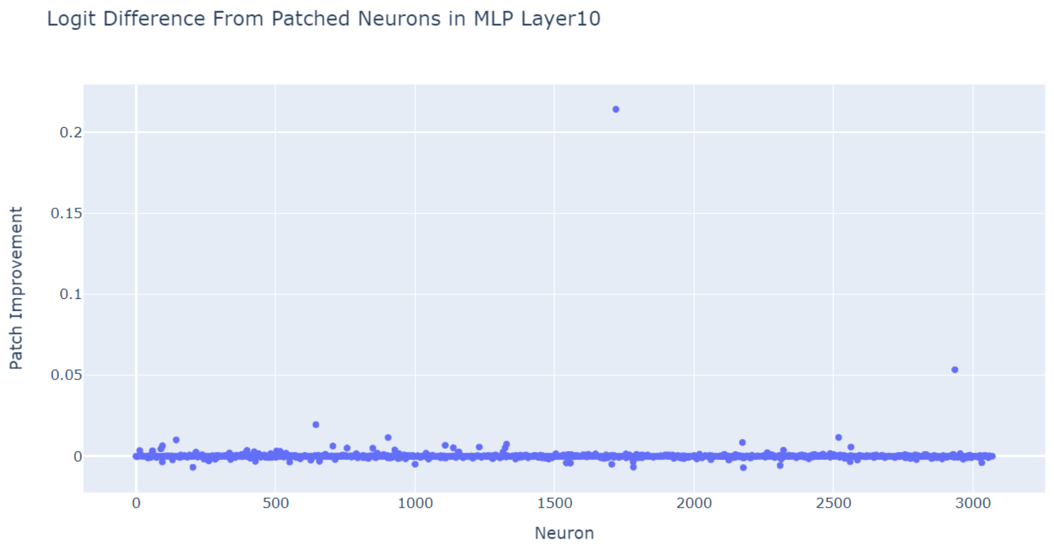
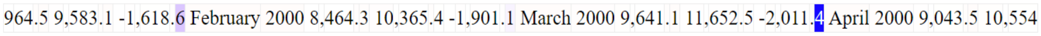


Figure 7 - Activation Patching by Neuron for L10. Neuron 1721 stand out as having more than 0.2 patch improvement

Neuroscope supports these findings by showing its strong activations near number-type tokens, as seen in Figure 8.



964.5 9,583.1 -1,618.6 February 2000 8,464.3 10,365.4 -1,901.1 March 2000 9,641.1 11,652.5 -2,011.4 April 2000 9,043.5 10,554

Figure 8 - A max activating example from the dataset for L10, N1721
Reference: https://neuroscope.io/gpt2-small/10/1721.html

As this section can involve deep analysis, and we spent a little time looking at it to uncover only a few pieces of evidence, there is likely much more to find.

6. Numerical Sequences Among Other Words

Adam is 1 … Eve is 5

Given that “1 2 3 4” does not clearly show information movement between different types of tokens due to being comprised solely of numerical digit tokens, we ran these experiments through other prompts that contained these sequences within other types of tokens. For instance, we tested these experiments on "Adam is 1. Bob is 2. Claire is 3. Don is 4. Eve is". We denote number-only sequences such as “1 2 3 4” as “pure”, and we denote sequences interspersed with non-numbers such as “Adam is 1…” as “non-pure”. We looked at two types of these inputs: using names as non-numbers, and using single-token random words as non-numbers.

We found that “Adam is 1” and “<random word> is 1” mostly had similar results compared to the pure sequence. The main difference between them and the pure sequence was that their heads 7.11 and 8.11 are much darker than in the pure digit sequence. However, we did not consider these differences to be significant enough to include more detailed analysis and comparisons.

Thus, we decided to compare the rest of the prompt types with multi-prompt “<name> is 1” rather than using a <random word>, as sentences with <name> makes more semantic sense to a reader. To control for unwanted variations, we used the same names for all the prompts, varying only the numbers. On the repo, these results are in the notebook: “actv_patch_word_is_num / multi_Adamis1_circuit_small.ipynb”.

Activation Patching

Patching from the residual stream showed information movement from the last digit to the last token.

Attention Heads

Early Heads
We notice that Early Heads attend to previous tokens of exactly the same or similar types. For instance, Figure 9 shows digit tokens attend to digit tokens. Thus, we hypothesize that these are 'similarity detection' heads.

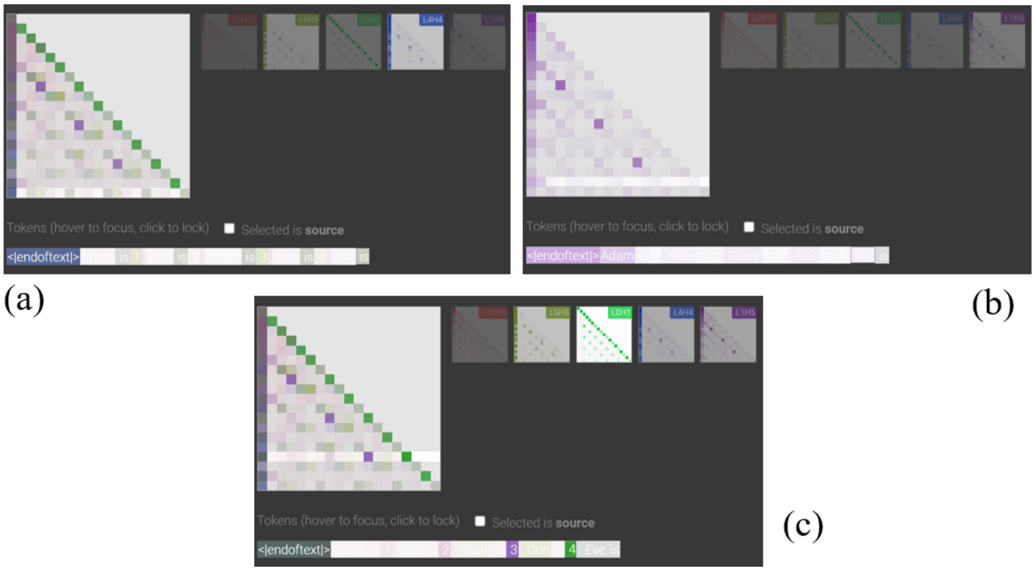


Figure 9 – Evidence of Early Heads where (a) "is" tokens attend to "is"tokens, (b) name tokens attend to name tokens, and (c) digit tokens attend to digit tokens

The brown highlights occur when most heads have the same type of query-key attention, as the color heads mix together into brown. Figure 8(b) shows only head 1.5, as it showed the strongest name-to-name attention, which overall is relatively weak.

Note that in contrast to the other top early heads which appear to be 'similar type detection' heads, 5.5 is an induction head, as shown in (Nanda, 2023c). In Figure 8(a), the olive highlights from a token to a token one position before the similar type are only from the induction head 5.5. Given that the input provides in-context patterns of "is <Number>", the induction heads are likely detecting then continuing this pattern.

Middle Heads

Both the middle and late heads attend from the final token to the number token. Unlike in IOI, where middle S-inhibition heads attended to the IO token to inhibit it, these middle heads appear to have stronger attention on numbers that were more recent in the input sequence. Thus, we hypothesize that they may be both boosting certain numbers and moving them to the final token.

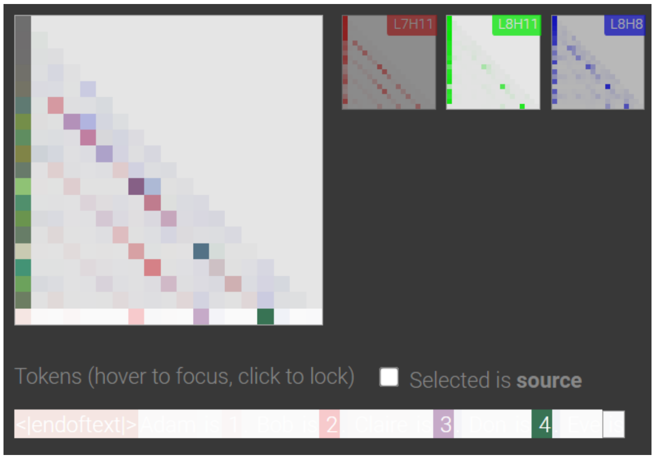
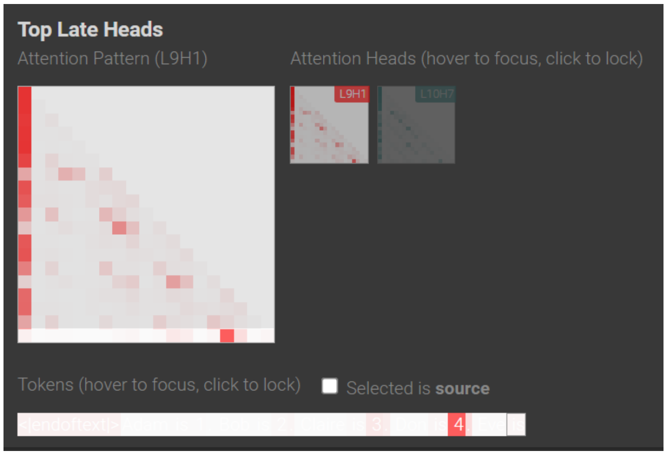


Figure 10 – Middle Heads for "Adam is 1…" prompts

However, there is an alternate explanation. All of these heads, in general, appear to attend to the last n-2 positions. Our pure sequence experiments with the same middle heads draw evidence contrary to this hypothesis. Yet another possibility is that each head attends more to a specific type of number, regardless of position. As we see in Figure 10 each digit aside from 3 is attended to by a different distinct (non-mixed) color of a head.

Late heads

Late head 9.1, which strongly and positively contributes according to the activation patching by heads heatmap in Figure 11, has much stronger attention to the most recent number relative to all other numbers. From our observations on the shading, it appears to attend even stronger to the most recent number than the previous heads in layers 7 and 8.



Top Late Heads

Attention Pattern (L9H1)

Attention Heads (hover to focus, click to lock)

Tokens (hover to focus, click to lock) □ Selected is **source**

<|endoftext|>Adam is 1. Bob is 2. Claire is 3. Don is 4. Eve is

Figure 11 – Late Heads for "Adam is 1…" prompts

<u>Refined Circuit Hypothesis</u>

Based on our findings from the results in previous sections, we further refine our hypothesis of how this circuit performs the task of sequence continuation.

- Early heads from token (to previous tokens of similar or exact same type) determined by V: We guess that these are 'similarity detection' heads
- Middle/Late heads from final token to number tokens determined by QK: these compose with early head outputs to copy the information about the duplicate token to final token to boost the attention paid from late (number mover) heads to the most recent number
- Late heads: given the unimportance of L10 and L11, we surmise these may be backup number heads, as backup name heads were mostly found in later layers in IOI (Wang et al., 2022). However, we require tests to provide evidence for this.

Note that what is "late vs middle" is arbitrarily drawn here. So unlike in IOI, it seems that 8.11 and 9.1 are doing similar things: they don't need to "inhibit" Mary, they just directly go to "boost" John. There may not be many differences between L7, L8 and L9.

To put it all together, we hypothesize that mid/late heads from final to previous number tokens compose with early head outputs, and then attend to the correct number AND copy that directly to the logits, using some way to boost attention based on how recent the number is. Then somehow, either by attention 9.1 or MLP, this copied head is associated (perhaps by key:value) (Meng et al., 2022) to the next number or sequence member.

Evidence that the heads are not just detecting the pattern “X is Y”

To check that these heads are not just detecting the general pattern of “X is Y”, but of “X is [number type]”, we require multiple “X is Y” prompts that are predictable in order to get correct and incorrect logit diff, and that are also corruptible that resulted in the incorrect token logit being greater than the correct token logit. Predictable corruption is non-trivial, so we only look at 1 prompt, which is “A is A.. (repeat 4 times). A is”, and the corruption switches the last to “B is”, which predicts B.

As shown in our notebook “AisA_circuit_small.ipynb”, we find that the plots result in very different trends and top heads, but that there are some heads in common. For instance, the middle heads 7.11 and 8.11 are found in both cases to attend from the end token to previous tokens, indicating they may have a more general purpose than being specific for sequence circuits. Additionally, we did not find any of the same ‘similarity detection’ early heads. Thus, this approach allows us to distinguish between heads specific to numbers, and heads involved in information movement in general. However, we are aware this is not a thorough experiment as it only has one input, and more tests should be done to rule out this scenario.

One is 1

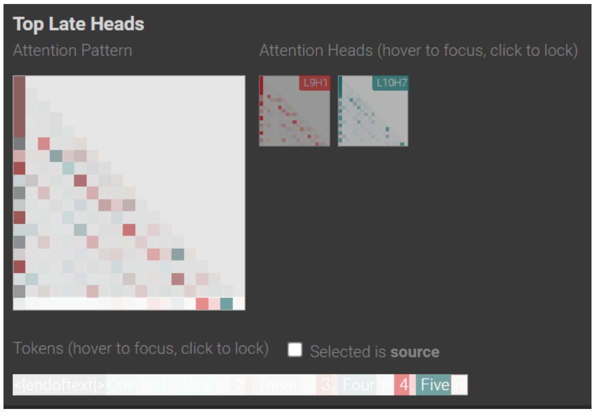

Top Late Heads
Attention Pattern
Attention Heads (hover to focus, click to lock)
L9H1
L10H7
Tokens (hover to focus, click to lock)
□ Selected is **source**
<|endoftext|>One is 1. Two is 2. Three is 3. Four is 4. Five is

Figure X - 10.7 attends to what’s before “is”, unlike in previous cases

An odd discovery is that the prompt “One is 1… Five” causes 10.7 to attend to “Five”, which is unusual as in previous cases it did not attend to what is before “is” that strongly. At first, we were led on a time-wasting red herring tangent to think that 10.7 had something to do with the number word, in competition with other heads that attend to the digit (like 9.1). However, we did not uncover more evidence of this. What’s more odd is that capitalization matters; “one is 1” does not have this pattern. But “1 is One” may have. The model seems to behave not on the one, but on the One. This is further evidence of cursed model behavior.

7. Mean-Ablation Circuit Analysis

This analysis uses code re-organized from (Mcdougall, 2023). For each sequence type, we run a dataset of multiple prompts through the model. This section has issues, and thus we would not include it in a more formal research paper, but in this writeup report we include errors, thought processes to double check methods, and negative results as they may help readers and beginners in this field to be aware of and avoid certain mistakes.

Mean ablation by a similar dataset that removes number value

Given that these sequence prompts have different structure than IOI, we investigate ways to choose a dataset to obtain mean activations from. In (Wang et al., 2022), a mean dataset that was close in structure to the main dataset (to measure scores from) was chosen. The paper states: "Mean-ablations remove the information that varies in the reference distribution (e.g. the value of the name outputted by a head) but will preserve constant information (e.g. the fact that a head is outputting a name)."

To avoid the mean dataset containing samples from the main dataset, we choose to split the datasets into two non-overlapping sets. For instance, the digits sequence dataset contains sequences 1 to 10 for the dataset to measure scores from, and the mean ablation dataset for it contains sequences 11 to 20. This way, information that the head is outputting a number is preserved, but the exact value is removed. However, the IOI paper mainly tried to remove information about selecting the correct IO; in our case, that appears to correspond to the mean dataset containing "sequential information". Therefore, our split datasets would not perform the removal correctly.

But the IOI paper also states zero ablation can lead to noisy results; therefore, we also seek to choose not to use a mean ablation from a dataset that was too dissimilar to the main dataset. We acknowledge that this choice of a mean dataset still has issues (eg, these sequences are still increasing by one, and perhaps it would be better to replace the numbers by some other non-increase-by-one predictable number sequence instead), so future work can perform more rigorous experiments to address them. Additionally, as we did not allocate enough time to figure out which token positions each head moves to during this study, we choose to keep all the sequence positions for the circuit heads we do not ablate. Information from future work about head functionality may better categorize these heads and their connective topology.

To check that our re-arrangement of the circuit score code is working correctly, we test the code on a circuit that contains every head. This results in scores identical to the full circuit. Additionally, to check that our circuit did not achieve high scores on every input (otherwise it would not be specifically geared towards continuing sequences), we test IOI prompts on it. This results in low scores. Finally, to check that each of our sequence prompts did not activate highly on just any circuit, we plot histograms of scores obtained from running the sequence prompts on randomly chosen circuits. The distribution shows that it is unlikely the scores come from random chance (N=20). We note that the scores appear to differ slightly each new restarted run, as there is some randomness involved. Thus, we focus on the relative comparison of each sub-circuit to the full circuit.

	Digits	Words	Months
Full	4.6237	3.4230	6.8312
Top 10 Pos	3.0229	3.0253	6.0312
L0 to L9	2.2351	2.8218	6.0974
Only 9.1	1.5308	2.1130	4.1479
Random	$\mu=0.9, \sigma=0.4$	$\mu=0.1, \sigma=0.4$	$\mu=1.4, \sigma=0.8$

Table 2a – Comparison of Average Logit Difference Scores for prompts (cols) run on different mean-ablated circuits (rows).

This would achieve good performance if the mean dataset was legitimate. However, it is still plagued by several issues: there are too few samples for numwords and months. Days of the week did not have enough samples to split, and we did not run on alphabet.

Note that the "Top 10 Pos" heads are from the digits column in Table 1. Out of curiosity, we also run this on "All Top 10", which include both positive and negative heads in the top 10; these are given in the digits column in Appendix A.

All Top 10	2.7761	2.9044	6.2434

Mean ablation by the same dataset

We find that using mean ablation that uses the same dataset achieves very high scores, albeit these scores are largely inflated by information that was not removed, so we do not consider these results to be very legitimate. We include this section just to show how much inflationary effect these results have, and we mostly run these results out of curiosity, like throwing various objects in a blender to see what would happen. A difference from before is that the sequence position they are not ablated on is just <end>.

	Digits	Words	Months	Days	Alphabet
Full	5.1332	2.6402	6.1420	2.4428	2.2813
Top 10	4.0804	2.6489	6.6191	2.4841	1.5980
L0 to L9	4.7670	2.3078	6.0295	2.1302	1.1679
Only 9.1	2.4358	1.5905	4.7845	1.6164	1.2870

Table 2b – Comparison of Average Logit Difference Scores for prompts (cols) run on different mean-ablated circuits (rows)

Like the models we study, we seek to not confuse the map for the territory; to not just memorize, but to generalize well. This principle is useful to gradually and carefully check for mistakes, and to not overshoot with erroneous conclusions.

8. Sub-Circuit Extraction and Surgery

This section is only done out of fun curiosity, but does not yield rigorous results. Our investigations with logit lens (nostalgebraist, 2020), found in the vector addition folder, showed that the model obtains the correct answer at L9. Thus, we experiment with a crude method to obtain new models by removing layers- either entire blocks, just attention, or just MLP- and stitching together the remaining layers. These experiments are found in the folder "extract_models". Some of the observations that include:

- Using only 9.1 or MLP_9 does not produce a model that performs sequence continuation; the previous layers are still important, likely as a way to transform the embeddings to a form that allows it to perform "sequence continuation"
- Omitting L8, attn9, L10 to L11 allows the model to even better on "1 2 3 4", and still do decently well on analogous sequences
- Other types of slice and stitchings allow it to still do well on other sequence types

What happens if the only attention heads we keep are the top attn heads L0 to L7?

9. Summary of Results

Main Positive Results:

- Analogous consecutive sequences share attention heads, and perhaps a circuit; even if there are issues with the choice of ablation for ablation experiments, the finding that they share attention heads suggest a commonality between them
- We discover attention heads that appear to respond to moving information regarding digits and months; these may be "number detector" and "number mover" heads
- There exist neurons, albeit polysemantic, that activate on numbers, rankings, months, and other analogous sequences

Other Takeaways for GPT-2 Small on sequence continuation:

- Token "Next" is highly ranked after the correct token for many sequence prompts
- Many prompts containing only two consecutive members of a sequence as the last two tokens will output the next member correctly
- The model is able to continue sequences even when interspersed with other tokens (albeit this may require an even spacing?)
- Capitalization has strange effects in some cases; "One is 1" has 10.7 with a strong negative effect, while "one is 1" does not

Main Negative Results:

- We did not locate an entire circuit with good faithfulness (with scores close to the full circuit)
- We did not find a clear way to get from digits to other sequences through latent space addition (a lot of time was spent on this, but it is beyond the scope of this project)

10.Future Work and Conclusion

In conclusion, by straightforwardly applying fundamental interpretability techniques, we find that consecutive sequences appear to share common activation patterns; whether or not this is an "abstract circuit" that others build on top of requires further testing. We also seek to perform validation on early head functionality.

Given that we can represent latent spaces using neurons as bases and vectors in this latent space are linear combinations of neurons, we can say that some of these vectors are features, and thus features correspond to directions. Using techniques from (Merullo et al., 2023), we attempt to find a path, through vector addition, to go from digits to numerical words, or a general "next" direction to go from a vector to its next element in a sequence. These paths were not found during this study, implying that the problem may be more complex than simple vector addition, so this work may be continued later.

Some more prompts to test include: Sequences of different lengths (incr just attends to last 2) decreasing sequences, sequences that depend on attending further back (eg. fibonacci), and interspersing non-numbers in less structured patterns than "X is Y". Many of these prompts were not tested as GPT-2 Small was incapable of predicting their next member correctly.

Other techniques to apply include ROME (Meng et al., 2022), ACDC (Conmy et al., 2023), and Neuron to Graph (Foote et al., 2023).

11. References

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., & Garriga-Alonso, A. (2023). Towards automated circuit discovery for mechanistic interpretability. arXiv preprint arXiv:2304.14997.

Foote, A., Nanda, N., Kran, E., Konstas, I., Cohen, S., & Barez, F. (2023). Neuron to Graph: Interpreting Language Model Neurons at Scale. arXiv preprint arXiv:2305.19911.

Mcdougall, C. (2023). ARENA_2.0. https://arena-ch1-transformers.streamlit.app/[1.3]_Indirect_Object_Identification

Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in GPT. Advances in Neural Information Processing Systems, 35, 17359-17372.

Merullo, J., Eickhoff, C., & Pavlick, E. (2023). Language Models Implement Simple Word2Vec-style Vector Arithmetic. arXiv preprint arXiv:2305.16130.

Miller, J., Neo, C. (2023) We Found An Neuron in GPT-2 https://www.lesswrong.com/posts/cgqh99SHsCv3jJYDS/we-found-an-neuron-in-gpt-2

Nanda, N. (2023a). 200 Concrete Problems In Interpretability Spreadsheet. https://docs.google.com/spreadsheets/d/1oOdrQ80jDK-aGn-EVdDt3dg65GhmzrvBWzJ6MUZB8n4/edit#gid=0

Nanda, N. (2023b). Exploratory Analysis Demo Notebook. https://colab.research.google.com/github/neelnanda-io/TransformerLens/blob/main/demos/Exploratory_Analysis_Demo.ipynb

Nanda, N. (2023c). Induction Mosaic. https://www.neelnanda.io/mosaic

Nanda, N. (2023d). TransformerLens. https://github.com/neelnanda-io/TransformerLens

nostalgebraist. (2020). interpreting GPT: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., & Steinhardt, J. (2022). Interpretability in the Wild: A Circuit for Indirect Object Identification in GPT-2 Small. https://arxiv.org/pdf/2211.00593.pdf

12. APPENDIX A– Top Heads Results (Positive and Negative)

Digits	NumWords
9.1: 0.21	9.1: 0.30
0.10: -0.14	0.10: -0.07
5.5: -0.10	5.5: -0.06
7.11: -0.08	0.1: 0.05
7.10: 0.06	8.8: 0.05
4.4: -0.06	7.10: 0.04
10.7: 0.04	6.1: 0.04
8.8: 0.04	6.10: -0.03
0.1: 0.03	3.0: -0.03
3.0: -0.03	0.5: 0.03
N/A	6.1, 6.10, 0.5

Table 1 - Top Attention heads and normalized patched logit difference for digit and number word input types. Last row: heads of that column's top heads that differ from the digit's top heads

13. APPENDIX B– Postponed Project Plan

One idea we sought to investigate further, but did not have time, was automating analysis (which is likely already an endeavor several are tackling). By using the current definitions of head functions, it may be possible to automate a pipeline of the analysis of identifying head functions and placing them in a possible circuit. One way this can be done is by checking attention patterns of important heads found by activation patching to see what tokens attend to what tokens. For instance, a head with tokens that attends to the same token much more than others may be considered a "duplicate detector", and those that attend to the "same type" may be considered a "<type> detector". The "semantic similarity" of tokens in terms of a <type> may be measured by the cosine similarity of their token embeddings, or by asking chatgpt after providing it with in-context examples of head functions or circuits. If a <type> appears in common to most of the tokens being strongly attended to, the algorithm would denote this as a <type> detector category. The composition of heads relative to others can also be taken into account to determine head function categories. Once a head is placed in a head function category, we can input a set of head function sets as a candidate circuit to minimal model ablation. Of course, this analysis automation may have obvious false positives or negatives that can be caught by a human researcher's checks. Thus, we only regard it as a starting point.