Gradient Based Circuits Discovery

by Pablo Talavante

Goal

Have an easy to implement method that can provide insights when looking for circuits

Goal

Have an easy to implement method that can provide insights when looking for circuits

General method that tell us *where to look*

Hypothesis

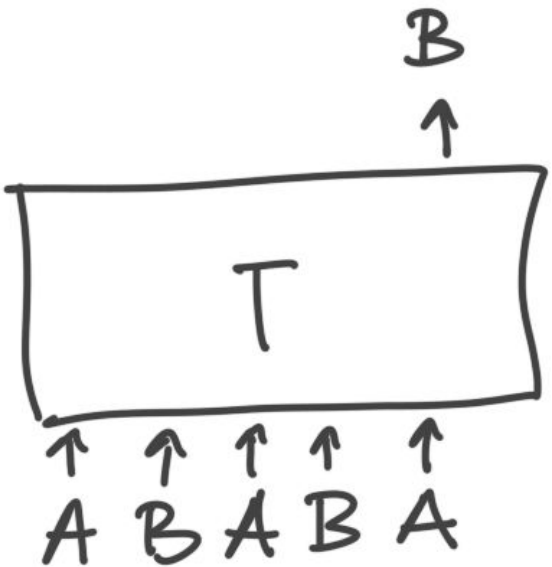
Given a transformer-based model **M**, and a task **T**, for which obtaining a correct answer **only** a (correctly working) circuit **C** is needed, the **gradients** in **M** for the weights involved in **C** will be maximal in a backward pass where the prediction is compared (via a loss function) to a **maximally wrong** answer.

Hypothesis

If we consider this scenario, to **detect** a circuit we would have to choose a **task** for which know **just C** is needed, get a prediction and get the **gradients** of the loss function comparing the prediction with the **maximally wrong answer**.

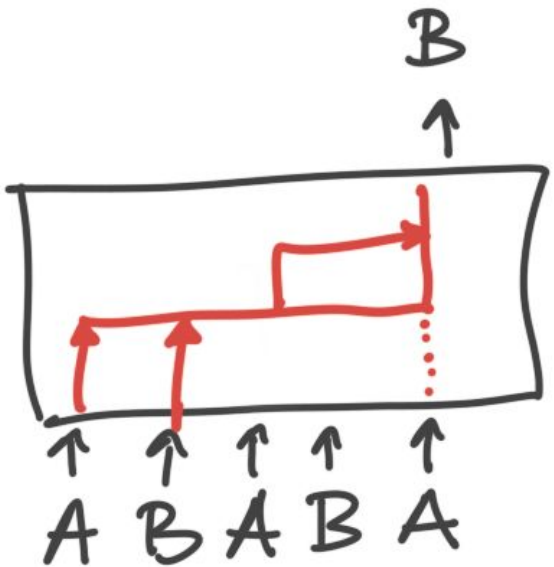
Eg.: induction circuits, IOI circuits

Task example



$\mathcal{L}(B, \bar{B})$

Task example



$\mathcal{L}(B, \bar{B})$

$\nabla \mathcal{L}(B, \bar{B})$

Task example

(as described in ACDC paper)

Task	Example Prompt	Output	Metric
1: IOI (Appendix E.2)	"When John and Mary went to the store, Mary gave a bottle of milk to"	"_John"	Logit difference

2: Docstring (Appendix G.1)	`def f(self, files, obj, state, size, shape, option):`
	`  """document string example`
	
	`  :param state: performance analysis`
	`  :param size: pattern design`
	`  :param`
		"_shape"	Logit difference

3: Greater-Than (Appendix F)	"The war lasted from 1517 to 15"	"18" or "19" or ... or "99"	Probability difference
4: tracr-xproportion (Appendix H.1)	`["a", "x", "b", "x"]`	`[0, 0.5, 0.33, 0.5]`	Mean Squared Error
5: tracr-reverse (Appendix H.2)	`[0, 3, 2, 1]`	`[1, 2, 3, 0]`	Mean Squared Error
6: Induction (Section 4.2)	"Vernon *Durs*ley and Petunia *Durs*"	"ley"	Negative log-probability

Maximally wrongness

How we define the maximally (optimally) wrong answer?

Possible definitions:

- most dissimilar token ?
- inverse of the logits ?
- logit difference ?
- semantics ?

Maximally wrongness

How we define the maximally (optimally) wrong answer?

Possible definitions:

- most dissimilar token ?
- inverse of the logits ?
- logit difference ?

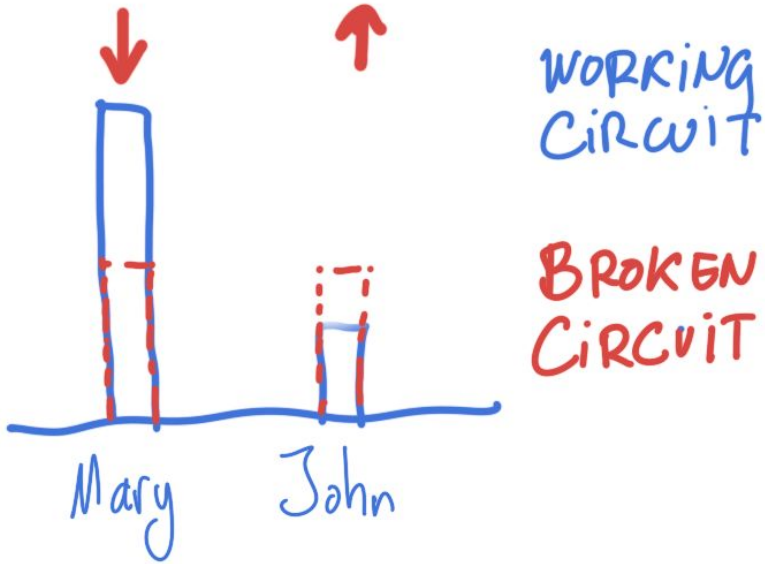
OPEN (?) QUESTION

Implementation (on GPT2-small)

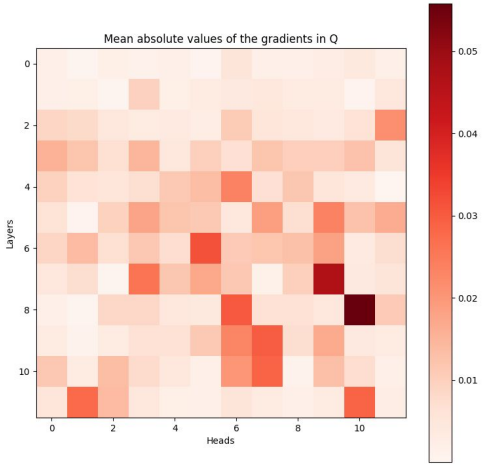
```
idx_john = t2idx[' John']
idx_mary = t2idx[' Mary']

text_A = 'When Mary and John went to the store, Mary gave a drink to'
text_B = 'When Mary and John went to the store, John gave a drink to'
input_ids_A, hidden_state_A, out_A = generate_sequence(text_A)
input_ids_B, hidden_state_B, out_B = generate_sequence(text_B)

loss = F.mse_loss(out_A[:, -1, idx_john],
                  out_A[:, -1, idx_mary]) + \
       F.mse_loss(out_B[:, -1, idx_john],
                  out_B[:, -1, idx_mary])
loss.backward()
```

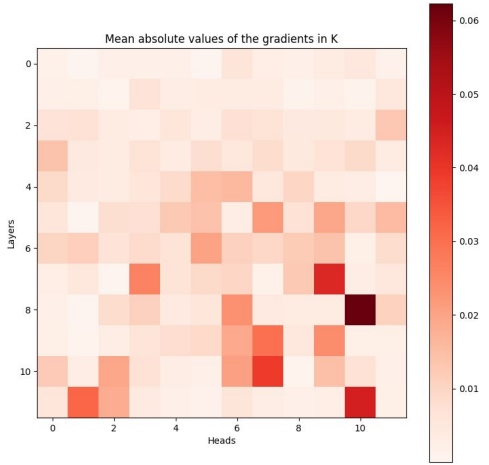
Implementation (on GPT2-small)

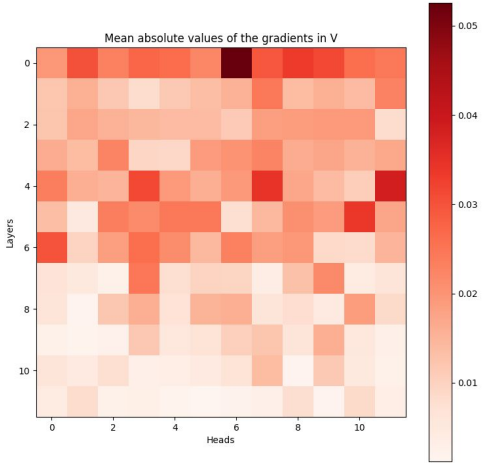
```
idx_john = t2idx[' John']
idx_mary = t2idx[' Mary']

text_A = 'When Mary and John went to the store, Mary gave a drink to'
text_B = 'When Mary and John went to the store, John gave a drink to'
input_ids_A, hidden_state_A, out_A = generate_sequence(text_A)
input_ids_B, hidden_state_B, out_B = generate_sequence(text_B)

loss = F.mse_loss(out_A[:, -1, idx_john],
                  out_A[:, -1, idx_mary]) + \
       F.mse_loss(out_B[:, -1, idx_john],
                  out_B[:, -1, idx_mary])
loss.backward()
```

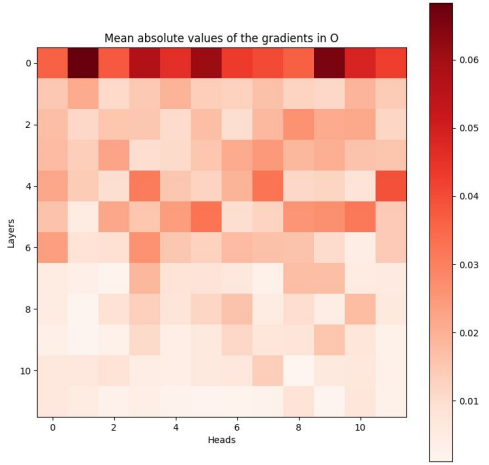


Results

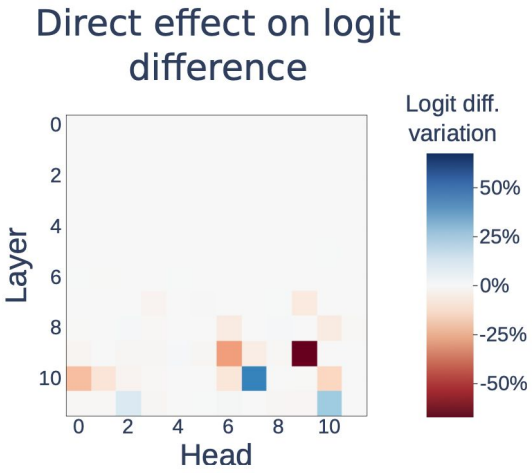

Mean absolute values of the gradients in Q
Layers
Heads
0
2
4
6
8
10
0.05
0.04
0.03
0.02
0.01

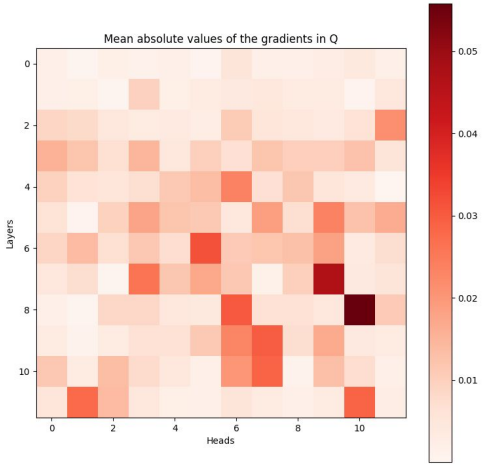

Mean absolute values of the gradients in K
Layers
Heads
0
2
4
6
8
10
0.06
0.05
0.04
0.03
0.02
0.01

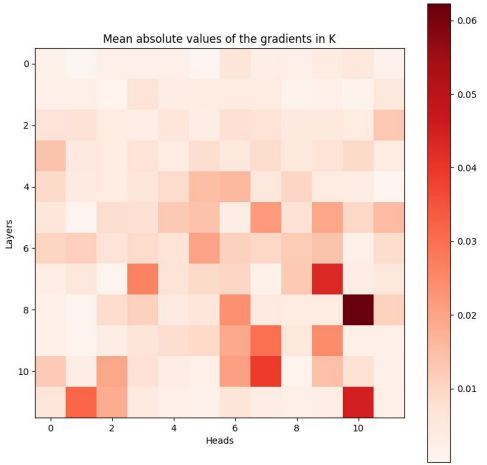

Mean absolute values of the gradients in V
Layers
Heads
0
2
4
6
8
10
0.05
0.04
0.03
0.02
0.01

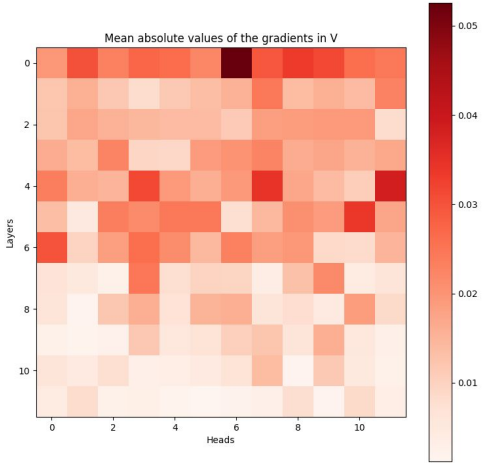

Mean absolute values of the gradients in O
Layers
Heads
0
2
4
6
8
10
0.06
0.05
0.04
0.03
0.02
0.01

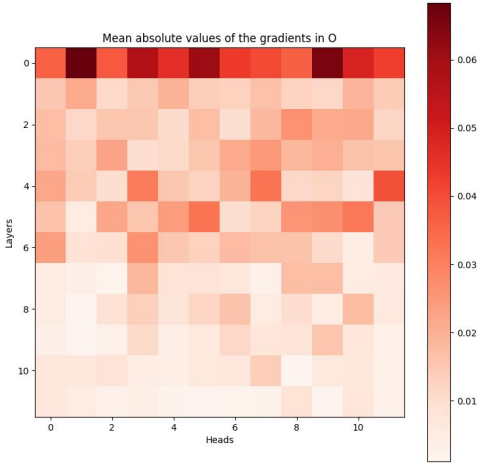
Results


Direct effect on logit difference
Logit diff. variation
50%
25%
0%
-25%
-50%
Layer
Head
0
2
4
6
8
10

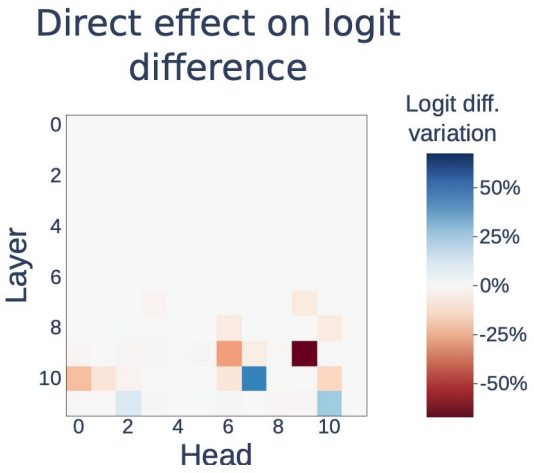

Mean absolute values of the gradients in Q
Layers
Heads

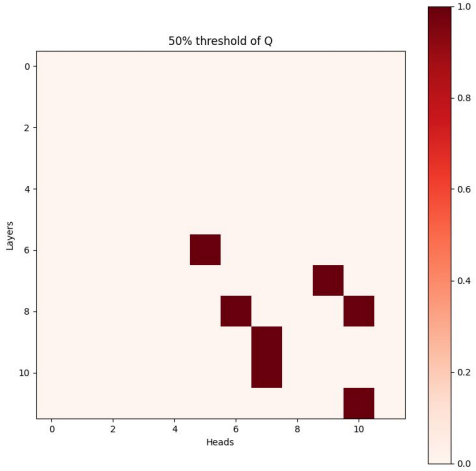

Mean absolute values of the gradients in K
Layers
Heads

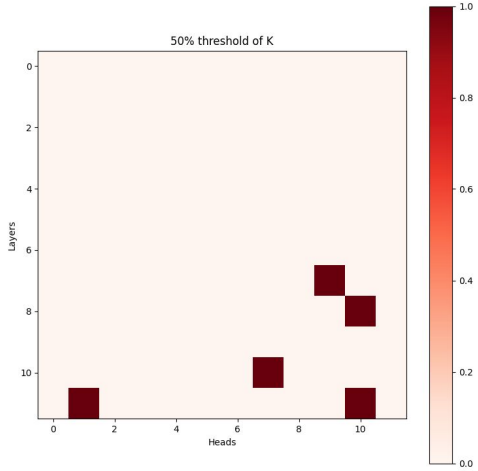

Mean absolute values of the gradients in V
Layers
Heads

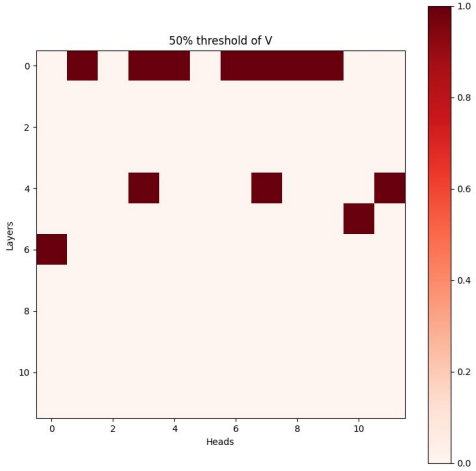

Mean absolute values of the gradients in O
Layers
Heads

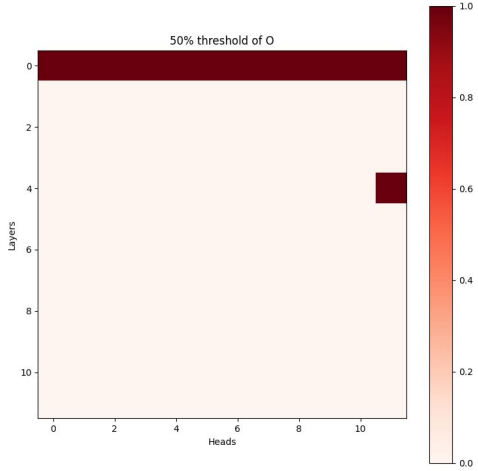
Results


Direct effect on logit difference
Logit diff. variation
50%
25%
0%
-25%
-50%
Layer
Head
0
2
4
6
8
10


50% threshold of Q
Layers
Heads


50% threshold of K
Layers
Heads


50% threshold of V
Layers
Heads


50% threshold of O
Layers
Heads

Pros

- There are no causal interventions
- Easy to implement
- Can give first insights on where to look

Cons

- There are no causal interventions
- Noisy outputs (due to circuit interference?)
- You work with gradients, which have a much larger dimensionality than activations

Thanks for your attention