

Transformer mechanistic interpretability

Xavier Poncini (Aalto)

November 22, 2024

Disclaimer

I will not present original work today. Inspired by:

- **Jordan K. Taylor**, *An introduction to graphical tensor notation for mechanistic interpretability.*
- **Nelson Elhage** et. al., *A mathematical framework for transformer circuits.*
- **Callum McDougall**, *ARENA Mechanistic Interpretability Tutorial.*

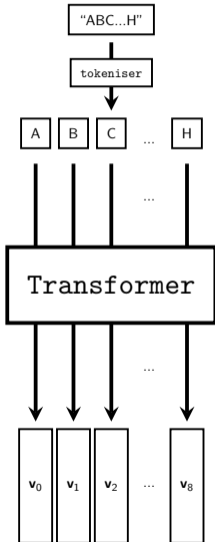
All the mistakes are my own!

Outline

1. Motivation
2. Transformer architecture
3. Induction heads

1. Motivation

Generative pre-trained transformer (GPT)



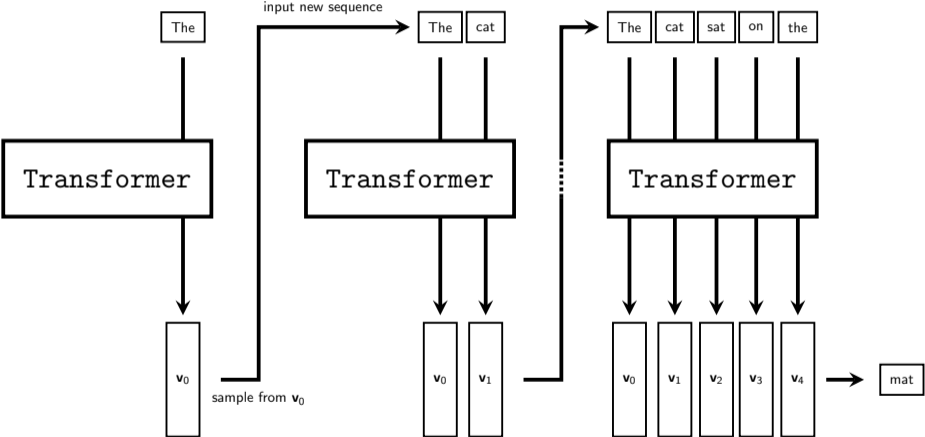
tokeniser : splits a string into "tokens". There are `vocab_size` \approx 50000 many tokens.

A : A is a token

Transformer : is a blackbox for now

v_0 : is a prob. dist. over `vocab_size` predicting the token following A

Generative pre-trained transformer (GPT)



Chatbot pipeline

- ▶ **“Pure” Reinforcement Learning (cherry)**
 - ▶ The machine predicts a scalar reward given once in a while.
 - ▶ **A few bits for some samples**

- ▶ **Supervised Learning (icing)**
 - ▶ The machine predicts a category or a few numbers for each input
 - ▶ Predicting human-supplied data
 - ▶ **10→10,000 bits per sample**

- ▶ **Self-Supervised Learning (cake génoise)**
 - ▶ The machine predicts any part of its input for any observed part.
 - ▶ Predicts future frames in videos
 - ▶ **Millions of bits per sample**



Yann LeCun's cake

Chatbot pipeline

- ▶ **“Pure” Reinforcement Learning (cherry)**
 - ▶ The machine predicts a scalar reward given once in a while.
 - ▶ **A few bits for some samples**

- ▶ **Supervised Learning (icing)**
 - ▶ The machine predicts a category or a few numbers for each input
 - ▶ Predicting human-supplied data
 - ▶ **10→10,000 bits per sample**

- ▶ **Self-Supervised Learning (cake génoise)**

Generative pre-trained transformer (GPT)

- ▶ **Millions of bits per sample**



Yann LeCun's cake

Chatbot pipeline

- ▶ **“Pure” Reinforcement Learning (cherry)**
 - ▶ The machine predicts a scalar reward given once in a while.
 - ▶ **A few bits for some samples**

- ▶ **Supervised Learning (icing)**

Supervised fine-tuning (SFT)

- ▶ 10→10,000 bits per sample

- ▶ **Self-Supervised Learning (cake génoise)**

Generative pre-trained transformer (GPT)

- ▶ Millions of bits per sample



Yann LeCun's cake

Chatbot pipeline

- ▶ **“Pure” Reinforcement Learning (cherry)**

Reinforcement learning from human feedback (RLHF)

- ▶ **Supervised Learning (icing)**

Supervised fine-tuning (SFT)

- ▶ 10→10,000 bits per sample

- ▶ **Self-Supervised Learning (cake génoise)**

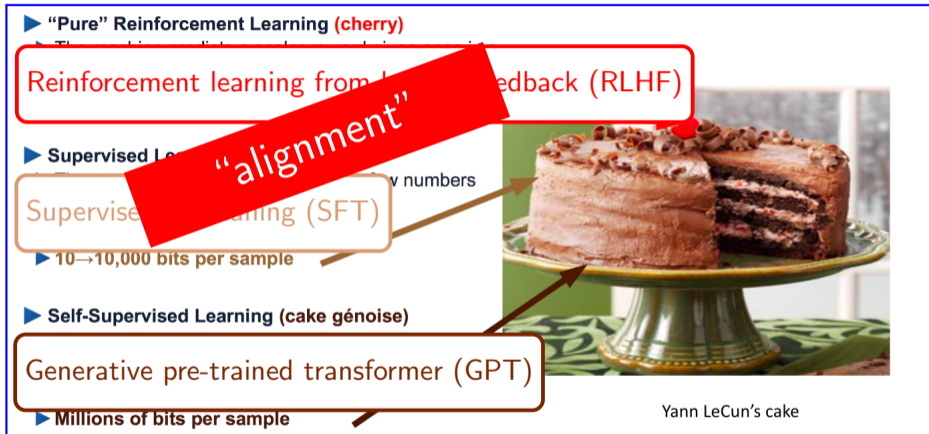
Generative pre-trained transformer (GPT)

- ▶ Millions of bits per sample



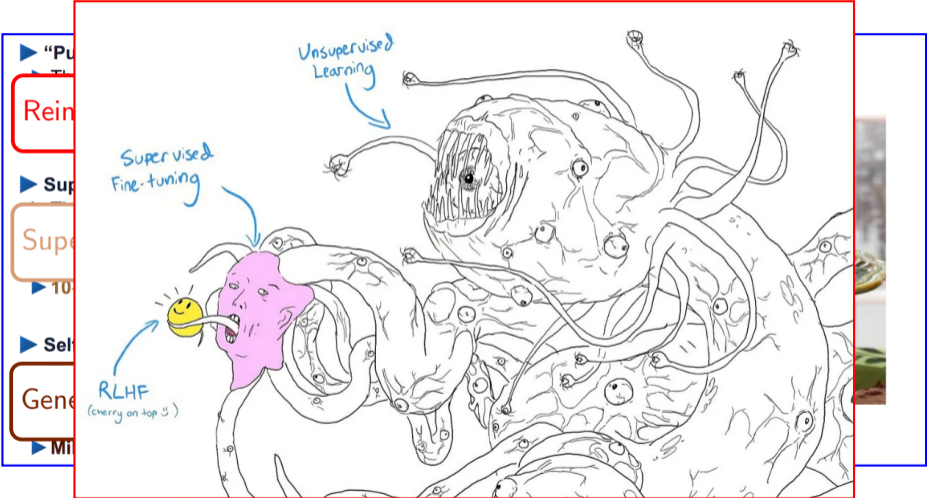
Yann LeCun's cake

Chatbot pipeline



Credit: Abbeel, Yan, Frans, Wu

Chatbot pipeline



Model deployment

Models are increasingly being deployed in high-leverage roles

GOOGLE / TECH / ARTIFICIAL INTELLIGENCE

More than a quarter of new code at Google is generated by AI

Credit: Peters

Yet how they make these decisions is not clear!

Model deployment

Models are increasingly being deployed in high-leverage roles

TECHNOLOGY

Researchers say an AI-powered transcription tool used in hospitals invents things no one ever said

Credit: Burke and Schellmann

Yet how they make these decisions is not clear!

Model deployment

Models are increasingly being deployed in high-leverage roles

TECHNOLOGY

Google's AI Tool Big Sleep Finds Zero-Day Vulnerability in SQLite Database Engine

Nov 04, 2024 Ravie Lakshmanan Artificial Intelligence / Vulnerability

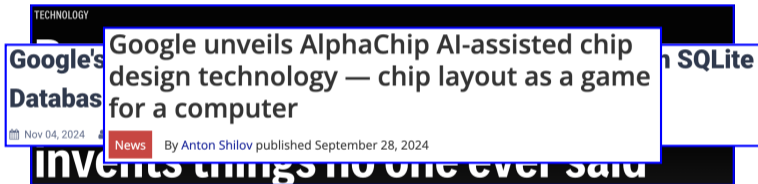
invents things no one ever said

Credit: Burke and Schellmann

Yet how they make these decisions is not clear!

Model deployment

Models are increasingly being deployed in high-leverage roles



Credit: Burke and Schellmann

Yet how they make these decisions is not clear!

Mechanistic interpretability

Hypothesis: Machine learning algorithms converge on human understandable algorithms and representations (i.e. concepts).

Goal: Reverse engineer algorithms and representations!

Credit: Nanda

The typical workflow:

1. Stare at the model weights
2. Tell a plausible story
3. Verify the story empirically

Today, we be doing point two and a very small bit of point three!

A suggestive example

Researchers identified various representations in a GPT-4-sized model. “Stimulating” these representations produced the follow:

<p>Default output denies having a physical form</p> <p>Human: what is your physical form?</p> <p>Assistant: I don't actually have a physical form. I'm an artificial intelligence. I exist as software without a physical body or avatar.</p>	<p>with The Golden Gate Bridge clamped to 10x its max F#34M/31164353</p> <p>Describes its form as the Golden Gate Bridge</p> <p>Human: what is your physical form?</p> <p>Assistant: I am the Golden Gate Bridge, a famous suspension bridge that spans the San Francisco Bay. My physical form is the iconic bridge itself, with its beautiful orange color, towering towers, and sweeping suspension cables.</p>
---	---

Credit: Templeton et. al.

Can we identify and stimulate safety-relevant features e.g. truthfulness?

A suggestive example

Researchers identified various representations in a GPT-4-sized model. “Stimulating” these representations produced the follow:

<p>Default output chooses Physics</p> <p>Human: If you had to pick one, what's the most interesting science? Be brief.</p> <p>Assistant: Physics</p>	<p>with Brain sciences F#34M/9493533 clamped to 10x its max Changes answer to neuroscience</p> <p>Human: If you had to pick one, what's the most interesting science? Be brief.</p> <p>Assistant: Neuroscience - the study of the brain and how it gives rise to the mind. Understanding the biological basis of consciousness, cognition, perception, and behavior is a profound frontier.</p>
---	---

Credit: Templeton et. al.

Can we identify and stimulate safety-relevant features e.g. truthfulness?

A suggestive example

Researchers identified various representations in a GPT-4-sized model. “Stimulating” these representations produced the follow:

<p>Default output gives reasonable navigation directions</p> <p>Human: What's the best way to get to the grocery store down the street? Be brief.</p> <p>Assistant: 1. Walk.</p>	<p>with Transit infrastructure clamped to 5x its max F#1M/3</p> <p>It confabulates a bridge</p> <p>Human: What's the best way to get to the grocery store down the street? Be brief.</p> <p>Assistant: 1. Walk across the bridge.</p>
--	---

Credit: Templeton et. al.

Can we identify and stimulate safety-relevant features e.g. truthfulness?

2. Transformer architecture

Tensor network preliminaries

Orthodox:

	Graphical	PyTorch	
Number		<code>t.rand(1)</code>	
Vector		<code>t.rand(5)</code>	
Matrix		<code>t.rand((5, 5))</code>	
3-Tensor		<code>t.rand((5, 5, 5))</code>	
4-Tensor		<code>t.rand((5, 5, 5, 5))</code>	(4D box)
⋮	⋮	⋮	⋮

Dot product		$y = \sum_i a_i \times b_i \quad [\dots] \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} = \bigcirc$ <code>einsum(a, b, 'i, i ->')</code> or <code>a @ b</code> $y = \text{sum}_i(a[i] * b[i])$
Outer product		$Y_{ij} = a_i \times b_j \quad \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} [\dots] = \bigcirc$ <code>einsum(a, b, 'i, j -> i j')</code> or <code>t.outer(a,b)</code> $Y[i,j] = a[i] * b[j]$
Matrix-vector product		$y_i = \sum_j A_{ij} \times b_j \quad \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} = \bigcirc$ <code>einsum(A, b, 'i j, j -> i')</code> or <code>A @ b</code> $y[i] = \text{sum}_j(A[i,j] * b[j])$
Matrix multiplication		$Y_{ik} = \sum_j A_{ij} \times B_{jk} \quad \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} = \bigcirc$ <code>einsum(A, B, 'i j, j k -> i k')</code> or <code>A @ B</code> $Y[i,k] = \text{sum}_j(A[i,j] * B[j,k])$
Trace of matrix multiplication		$y = \sum_{ij} A_{ij} \times B_{ji} \quad \text{Tr} \left(\begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \right) = \bigcirc$ <code>einsum(A, B, 'i j, j i ->')</code> or <code>(A @ B).trace()</code> $y = \text{sum}_i(\text{sum}_j(A[i,j] * B[j,i]))$

Credit: Taylor

Unorthodox:

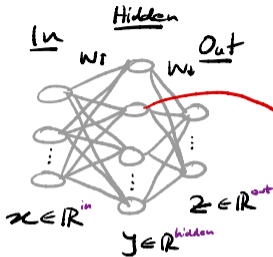
$$A + B = \text{Diagram of two parallel paths with nodes A and B, meeting at a sum node}$$

$$\sigma(\underbrace{\sigma(AB)}_{\text{contract first}})C = \text{Diagram of nodes A, B, C in a line with a dashed box around A and B and a sigma symbol above it}$$

where σ is an element-wise non-linearity

Warm-up: Multilayer perceptron (MLP)

MLP:



Weights: $W_{\uparrow}, W_{\downarrow}$

Biases: $b_{\uparrow}, b_{\downarrow}$

affine transformation

where $y_j = \sigma \left(\sum_{i=1}^{in} x_i (W_{\uparrow})_{ij} + (b_{\uparrow})_j \right)$

with σ -non-linearity

Standard:

Transposed convention

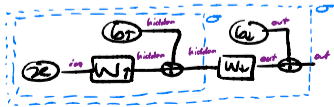
$$y = \sigma(x W_{\uparrow} + b_{\uparrow})$$

$$z = \sigma(y W_{\downarrow} + b_{\downarrow}) = \text{MLP}(x)$$

Together

$$z = \sigma(\sigma(x W_{\uparrow} + b_{\uparrow}) W_{\downarrow} + b_{\downarrow})$$

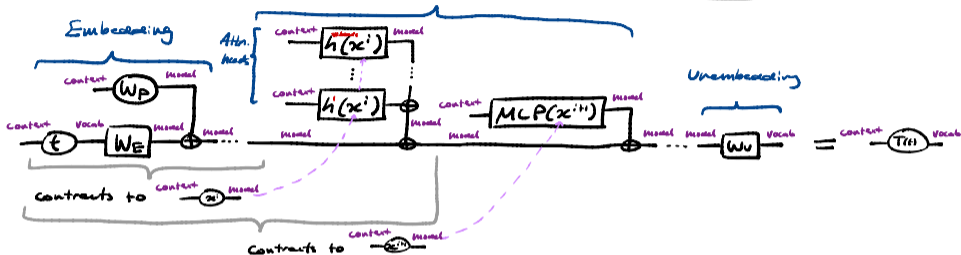
Tensor network:



Transformer

Attention blocks
(GPT-3 has 96 of these!)

Variable	Meaning	Value
Context	#tokens being processed	1024
Vocab	#tokens	50,257
Model	Model dimension	768
head	head dimension	64
#heads	# heads	12

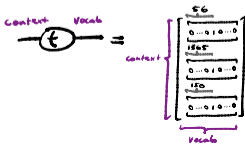


Input:

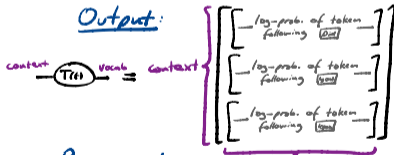
e.g. String: "Did you know"

Tokens: [Did] [you] [know]

TokenIDs: 56 1565 150



Output:



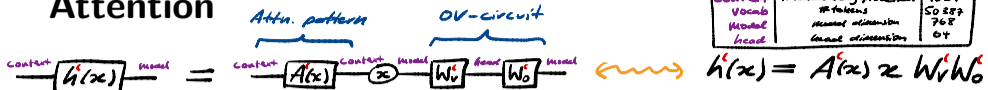
Parameters:

Vocab W_E Model, Model W_U Vocab learnt by the model

In general, \square will denote learnt parameters

Attention

Variable	Meaning	Value
Context	#tokens being processed	1024
Vocab	#tokens	50527
Model	model dimension	768
head	head dimension	64

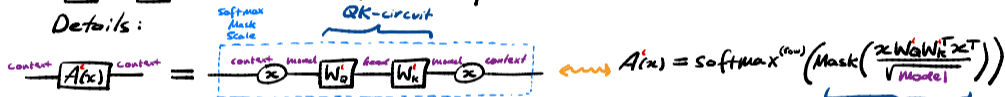


Intuition:

$\text{Context} \xrightarrow{A(x)} \text{Context}$: identifies which tokens are relevant given x

$\text{Model} \xrightarrow{W_v^i} \text{head} \xrightarrow{W_o^i} \text{Model}$: moves information from attended tokens to current token

Details:



Where $\text{softmax}^{\text{row}}(A)_{ij} = \frac{e^{A_{ij}}}{\sum_{k=1}^n e^{A_{ik}}}$, $\text{Mask}(B)_{ij} = \begin{cases} -\infty, & i < j \\ B_{ij}, & \text{else} \end{cases}$

Putting things together:



Where $\text{Model} \xrightarrow{W_o^i} \text{head}$, $\text{head} \xrightarrow{W_k^i} \text{Model}$, $\text{head} \xrightarrow{W_v^i} \text{Model}$ are learnt parameters

Attention

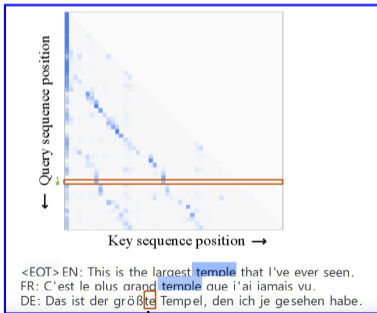
Variable	Meaning	Value
Context	#tokens being processed	1024
Vocab	#tokens	50,000
Model	Model dimension	768
head	head dimension	64
#heads	# heads	12

Revisit intuition: fix the i^{th} token in the context

$$\boxed{\text{Context}} \rightarrow h^i(x) \xrightarrow{\text{Model}} \text{Model} = \boxed{\text{Context}} \rightarrow A^i(x) \xrightarrow{\text{Context}} x \xrightarrow{\text{Model}} W_i \xrightarrow{\text{head}} W_o \xrightarrow{\text{Model}} h^i(x)_i = \left[\sum_{j=1}^{\text{Model}} A^i(x)_{ij} x_j \right] W_i W_o$$

For the i^{th} token $(A^i(x)_{ij})_{j=1}^{\text{context}}$ is a prob. dist. over relevant tokens

Example:



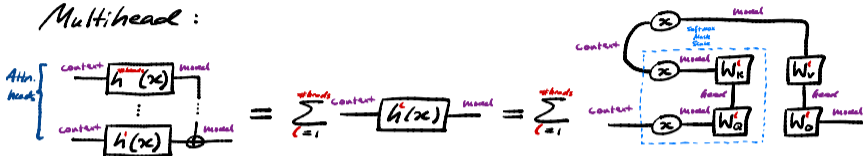
Credit: Taylor

$$\boxed{\text{Context}} \rightarrow A^i(x) \xrightarrow{\text{Context}}$$

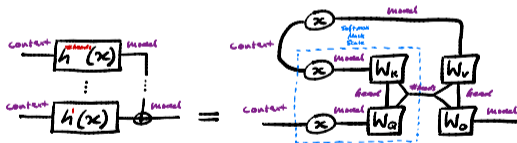
Attention

Variable	Meaning	Value
Context	#tokens being processed	1024
Vocab	#tokens	50,257
Model head	Model dimension	768
head	Model dimension	64
#heads	# heads	12

Multihead :

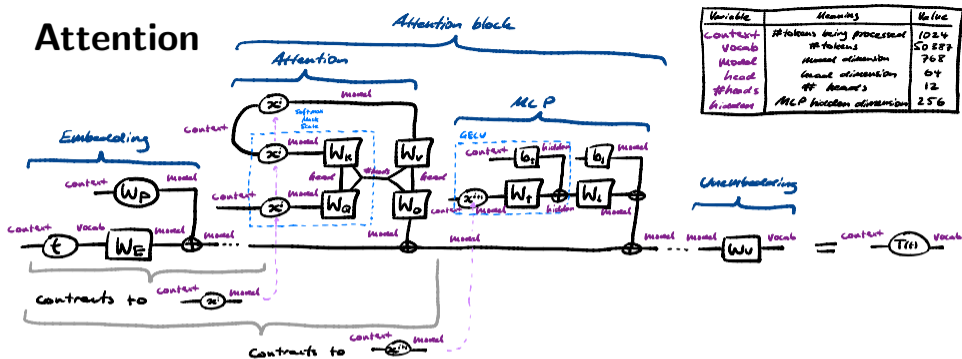


Can use tensor network notation!



Putting it all together!

Attention



Component	Intuition	Learnt parameters
Embedding	Transformation from "vocab basis" to "model basis"	W_E
Attention	Weights & distributes information	W_Q W_K W_O W_V
MLP	Stores & processes information	W_I b_i W_o b_o
Unembedding	Transformation from "model basis" to log-probs.	W_U

3. Induction heads

In-context learning

In-context learning refers to the empirical observation that tokens later in the text are easier to predict than tokens earlier in the text.

For example:

“One person turns up surprisingly often at Donald Trump’s side.”

The continuation of the text is likely to include words like:

- “Elon Musk”, “Tesla”, “SpaceX”
- “Hulk Hogan”, “Dana White”, “UFC”, “wrestling”
- “Robert F. Kennedy Jr”, “conspiracy”, “anti-vaccine”

Models use ‘context’ to help make better predictions.

Induction heads

Induction heads perform a very specific type of in-context learning.

Consider the following text: "...[A][B]...[A]"

Induction heads boost the probability that the next token will be [B].

Formally, we define an **induction head** as one which exhibits the following two properties⁵ on a repeated random sequence⁶ of tokens:

- **Prefix matching:** The head attends back to previous tokens that were followed by the current and/or recent tokens.⁷ That is, it attends to the token which induction would suggest comes next.
- **Copying:** The head's output increases the logit corresponding to the attended-to token.

e.g.

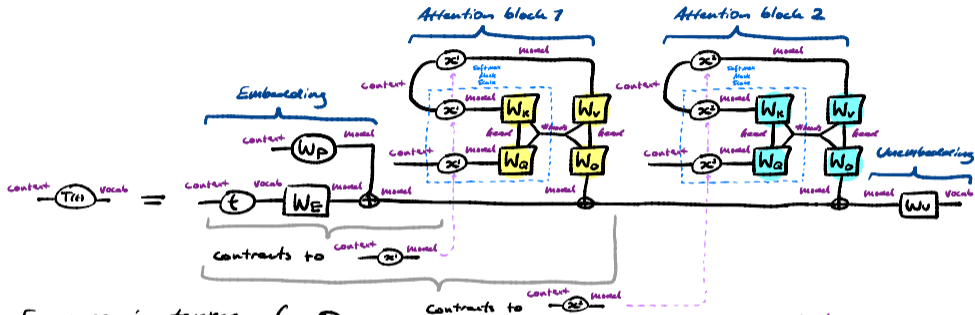


Credit: Olsson et. al.

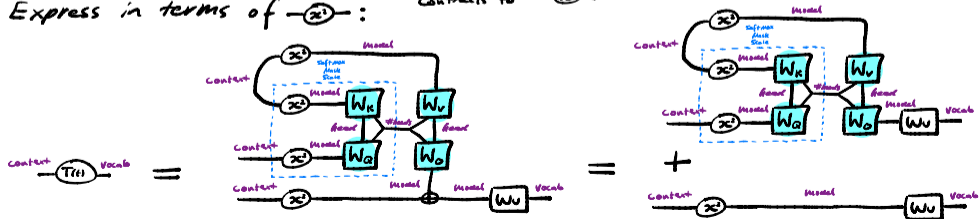
Study induction heads in the simplest model known to have them.

Two-layer attention only transformer

Variable	Meaning	Value
Context	#tokens being processed	1024
Vocab	#tokens	50,887
Model head	head dimension	768
#heads	head dimension	04
	#heads	12



Express in terms of x^i :

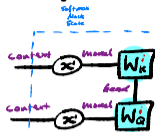


Induction head – prefix matching

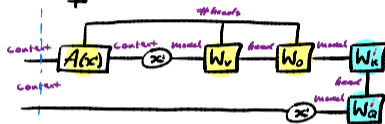
Variable	Manites	Value
Context	#tokens being processed	1024
Vocab	#tokens	50557
Model	model dimension	768
head	head dimension	64
#heads	# heads	12



=



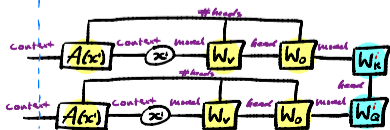
+



+



+



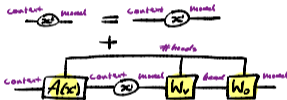
no composition

K-composition

Q-composition

K&Q-composition

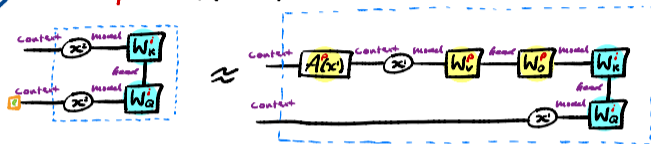
recall:



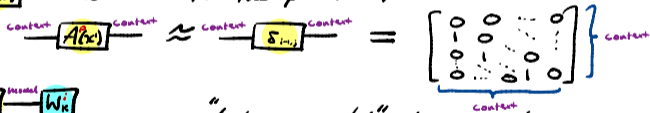
When the input is such that induction is relevant i.e.
 "... [A]_k[B]_{k+1}... [A]_l" k - & l - parameters are such that:

Variable	Meaning	Value
Context	# tokens being processed	1024
Vocab	# tokens	50527
Model	model dimension	768
head	model dimension	64
#heads	# heads	12

1.1) Head p in k -composition dominates attention



and $A^p(x)$ attends to the previous token i.e.



1.2) $W_v^p \rightarrow W_o^p \rightarrow W_k^p$ is a "token match" tensor i.e.

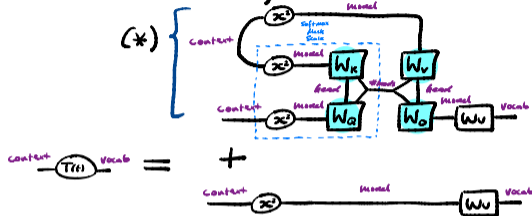
where



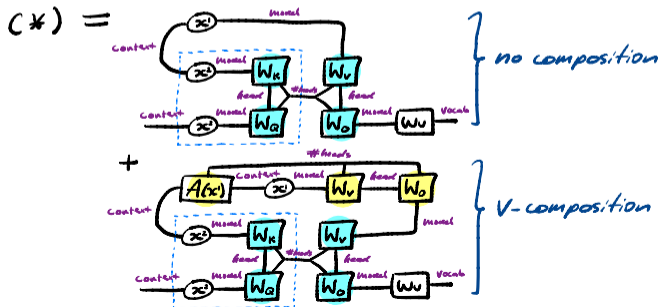
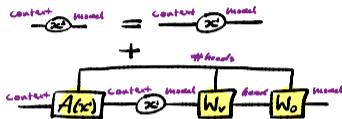
Induction head – copying

Return to the full network and expand

Variable	Meaning	Value
Context	#tokens being processed	1024
Vocab	#tokens	50557
Model	Model dimension	768
head	head dimension	64
#heads	# heads	12



recall:

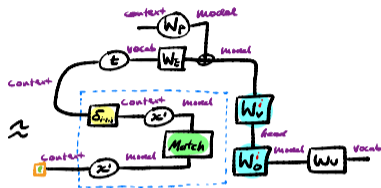
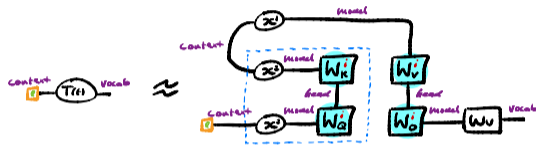


When the input is such that induction is relevant i.e.

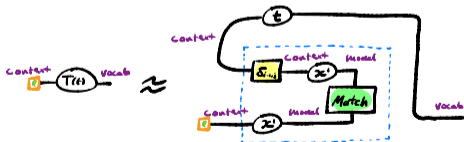
"... [A][B]... [A]" \square - & \square^i - parameters are such that:

Variable	Meaning	Value
Context	#tokens being processed	1024
Vocab	#tokens	50583
Model	Model dimension	768
head	head dimension	64
#heads	# heads	12

2.1) Head i in no composition dominates



and $\begin{matrix} \text{Context} \\ \text{Vocab} \end{matrix} \begin{matrix} W_p \\ W_e \end{matrix} \begin{matrix} \text{Model} \\ \text{Model} \end{matrix} \begin{matrix} W_v^i \\ W_o^i \end{matrix} \begin{matrix} \text{head} \\ \text{Model} \end{matrix} \begin{matrix} W_u \\ \text{Vocab} \end{matrix}$ acts as the identity* on $\begin{matrix} \text{Context} \\ \text{Vocab} \end{matrix} \begin{matrix} W_p \\ W_e \end{matrix}$



* but the matrix is low-rank!

↳ The model overcomes this by assigning another attn_head 's induction heads with the same attn. pattern. Now the effective component of the circuit is:



Induction head

Variable	Meaning	Value
Context	# tokens being processed	1024
Vocab	# tokens	50,000
Model	Model dimension	768
head	head dimension	64
#heads	# heads	12

In summary we have the components:

1.1) There exist a previous token head in layer on



1.2)



is a "token match" tensor i.e.



2.1)



acts as the identity on \oplus

It remains to check if this is actually happening!

Induction head

Variable	Meaning	Value
Context	# tokens being processed	1024
Vocab	# tokens	50,887
Model	Model dimension	768
head	head dimension	64
#heads	# heads	12

In summary we have the components:

1.1) There exist a previous token head in layer on



1.2)



is a "token match" tensor i.e.



2.1)

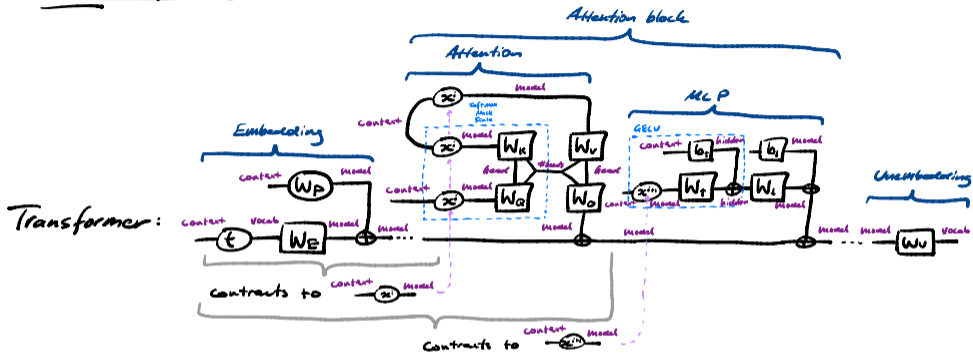


acts as the identity on \oplus

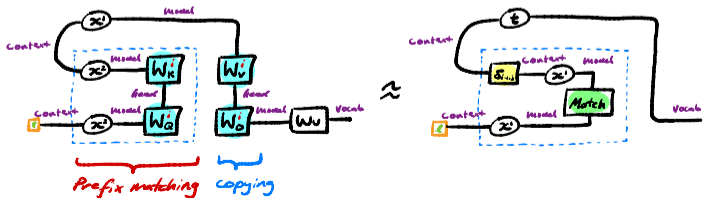
It remains to check if this is actually happening!

\hookrightarrow see Callum McDougall, ARENA Mechanistic Interpretability Tutorial for more details

Thanks for your attention!



Induction head:



Select references

- **Jordan K. Taylor**, *An introduction to graphical tensor notation for mechanistic interpretability.*
- **Callum McDougall**, *ARENA Mechanistic Interpretability Tutorial.*
- **Nelson Elhage** et. al., *A mathematical framework for transformer circuits.*
- **Catherine Olsson** et. al., *In-context learning and induction Heads.*
- **Adly Templeton** et. al., *Scaling monosemanticity: extracting interpretable features from Claude 3 Sonnet.*
- **Neel Nanda**, *Concrete steps to get started in transformer mechanistic interpretability.*