Foundations First: Understanding GPT-1

Jacob Danner

Opening

- Who here:
  - Has used a GPT style model at the API level?
  - Has trained, fine-tuned or done any sort of hands on programming with a neural network?
  - Is unsatisfied with the “black box” explanation of AI?

- Feel free to interrupt with any questions

Language Modeling

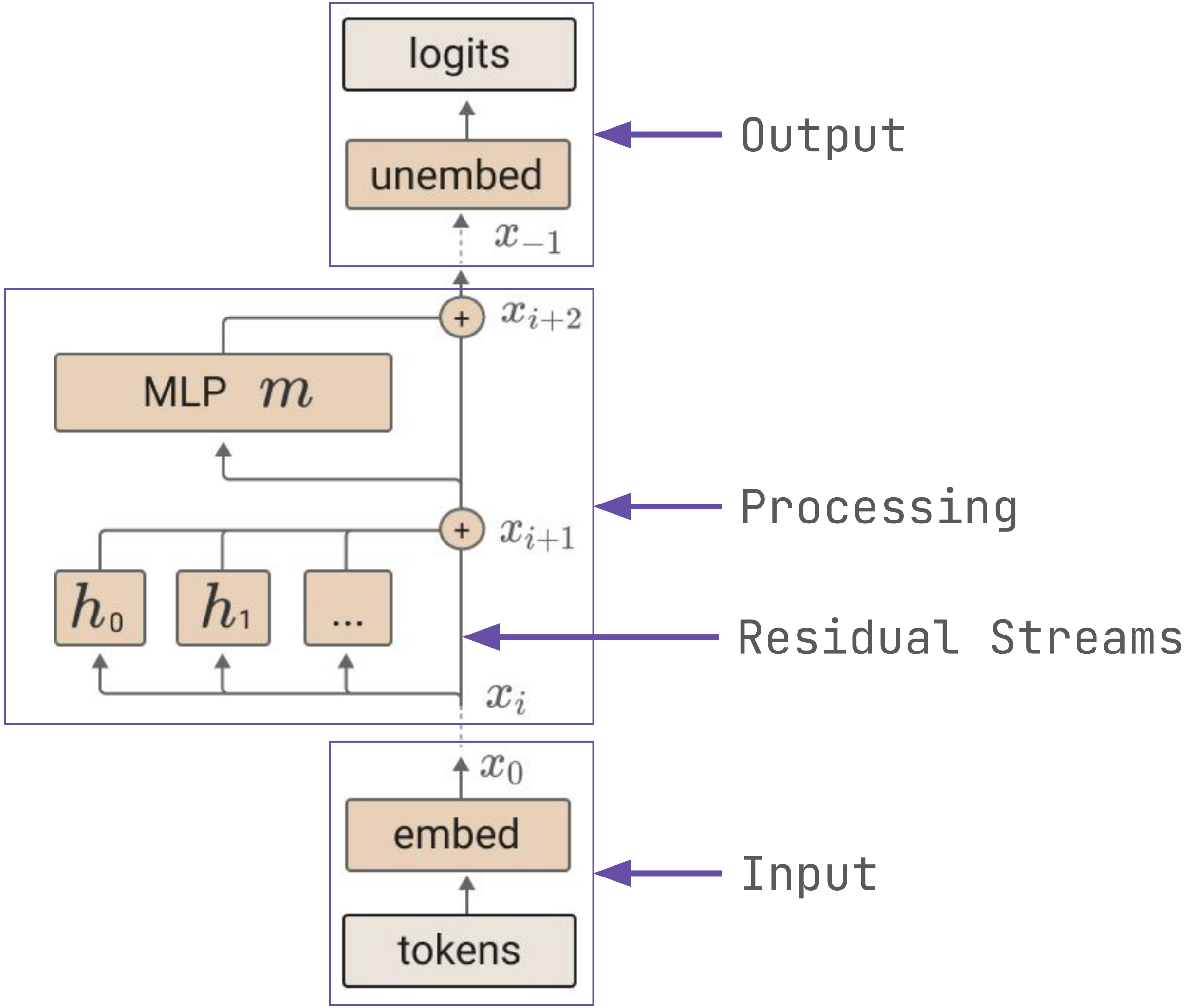
- **Defn**: A language model is a system that assigns probabilities to sequences of words, enabling it to generate, predict, or evaluate text.
- Old approaches:
  - N-gram models
  - Hidden Markov models
  - Feedforward neural networks
  - Recurrent neural networks
- *“Attention Is All You Need”* (2017, Google)
  - Encoder-Decoder Transformer
  - Translation

The GPT Family

- *"Improving Language Understanding by Generative Pre-Training"* (2018, OpenAI)
  - **G**enerative
  - **P**re-trained
  - **T**ransformer (decoder)
  - Specs:
    - 40 GB of text (~1 billion tokens)
    - 117 million parameters
- Descendents:
  - ChatGPT
  - Claude
  - Gemini
  - etc.

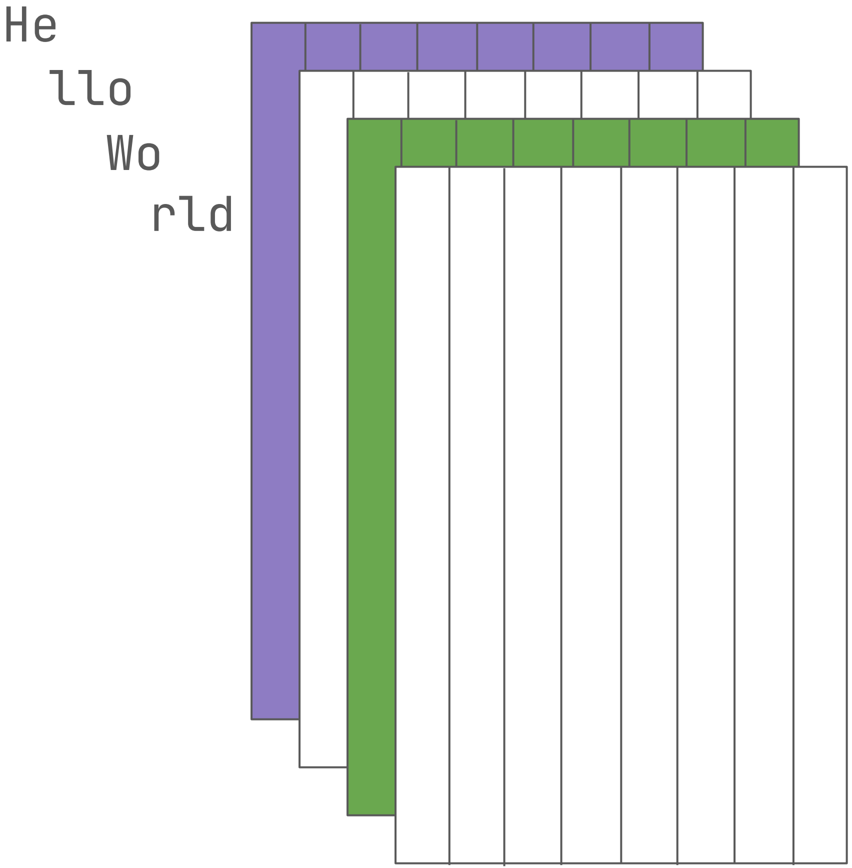
GPT: What to Know

- Input
  - Tokens
  - Embed
- Processing (transformer blocks)
  - Attention (contextualizing tokens)
    - QK
    - OV
  - MLP (refining token representations)
  - Residual stream (the information highway)
- Output
  - Unembed
  - Sampling


logits
unembed
x_{-1}
Output
x_{i+2}
MLP m
Processing
x_{i+1}
h_0
h_1
...
Residual Streams
x_i
x_0
embed
Input
tokens

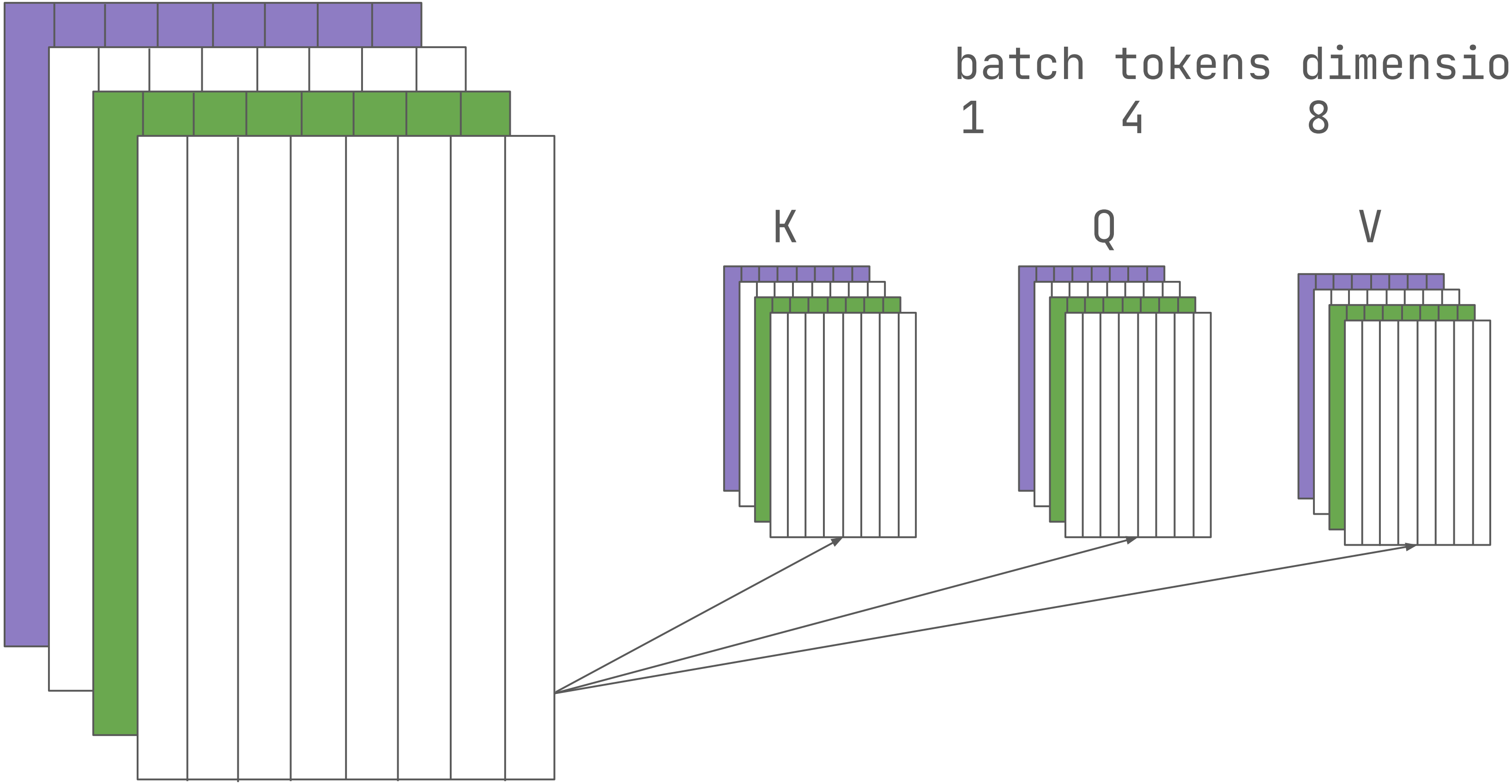
$$
\begin{aligned}
h_0 &= UW_e + W_p \\
h_l &= \mathtt{transformer\_block}(h_{l-1}) \forall i \in [1, n] \\
P(u) &= \mathtt{softmax}(h_n W_e^T)
\end{aligned}
$$

Initial Residual Streams



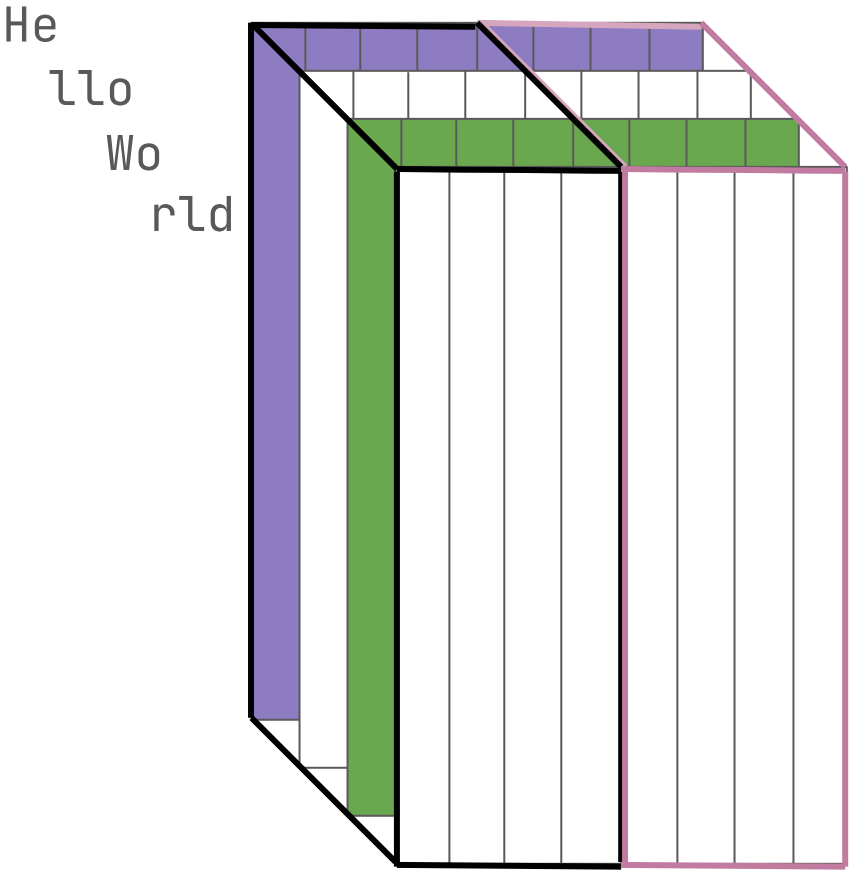
batch	tokens	dimension
1	4	8

Initial Residual Streams (K, Q, V)



batch	tokens	dimension
1	4	8

2 attention heads



batch	n_heads	tokens	dimension
1	2	4	4

Attention (looking at 1 head)

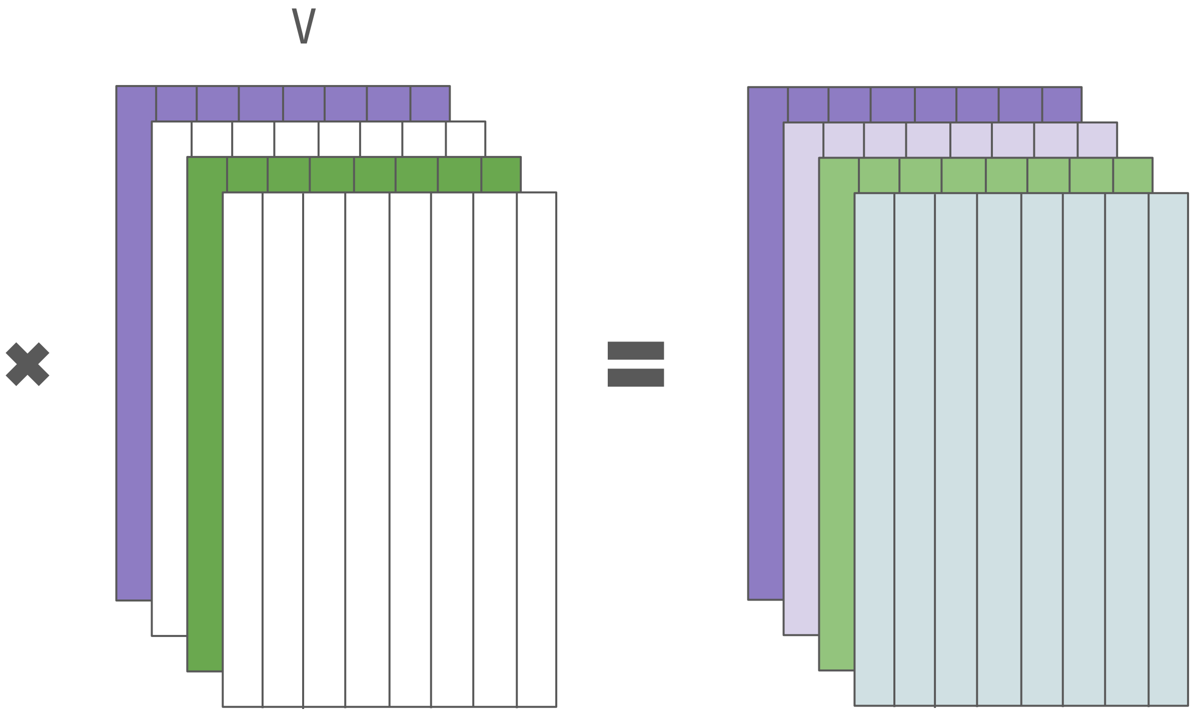
Q @ K^T

	He	llo	Wo	rld
He	1	-	-	-
llo	.7	.3	-	-
Wo	.2	.2	.6	-
rld	.2	.2	.2	.4

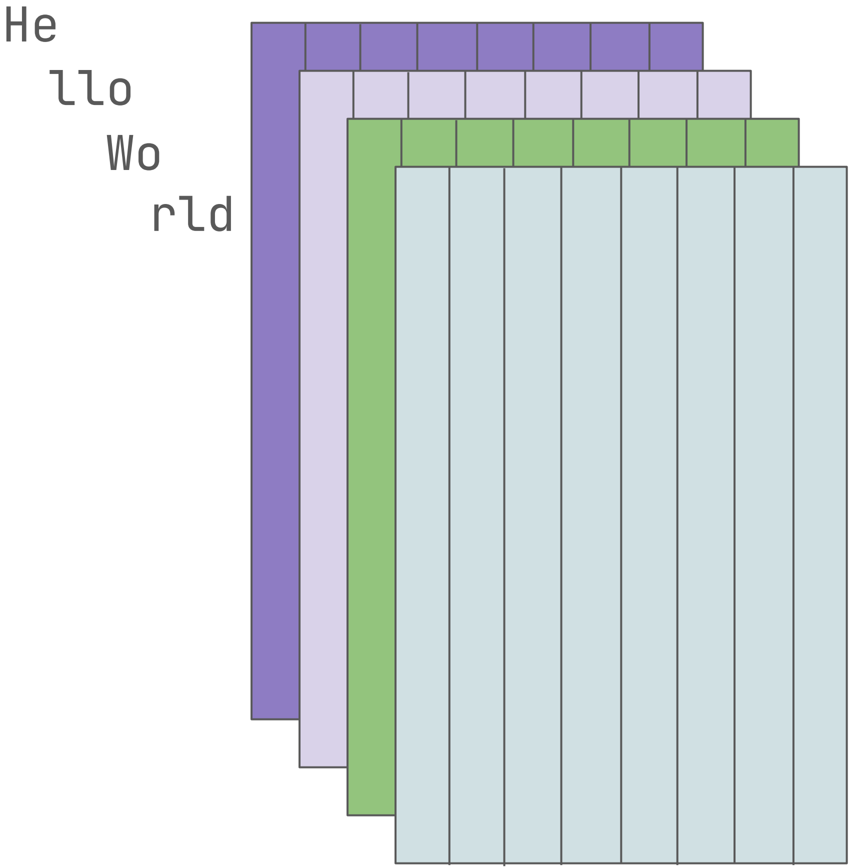
Attention (looking at 1 head)

Q @ K^T

	He	llo	Wo	rld
He	1	-	-	-
llo	.7	.3	-	-
Wo	.2	.2	.6	-
rld	.2	.2	.2	.4

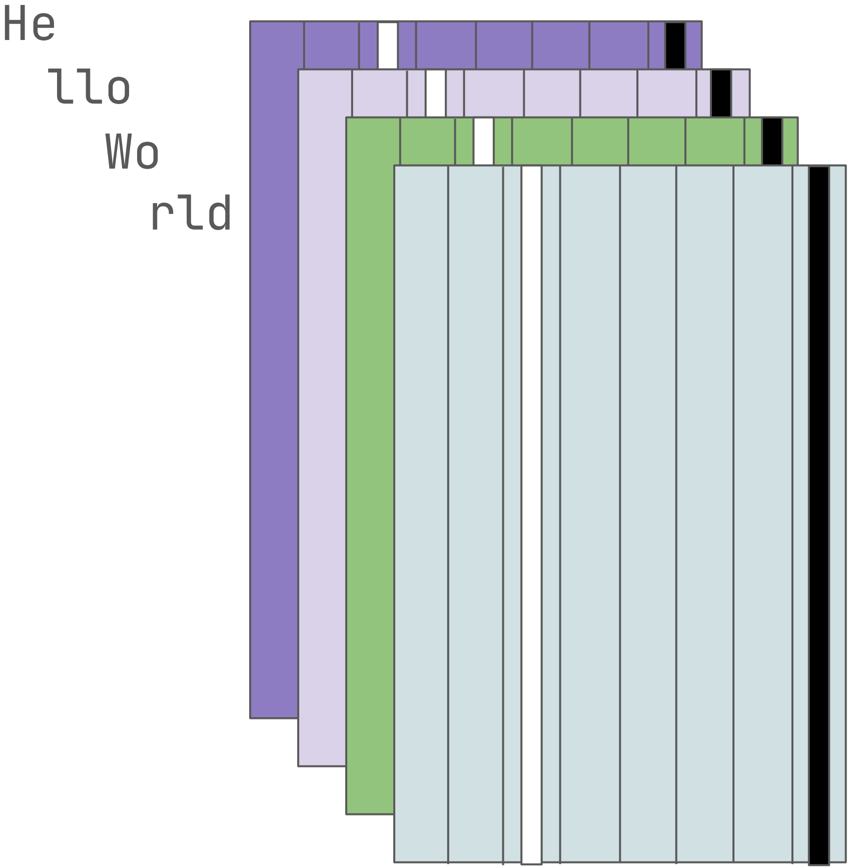


Residual Streams After Attention



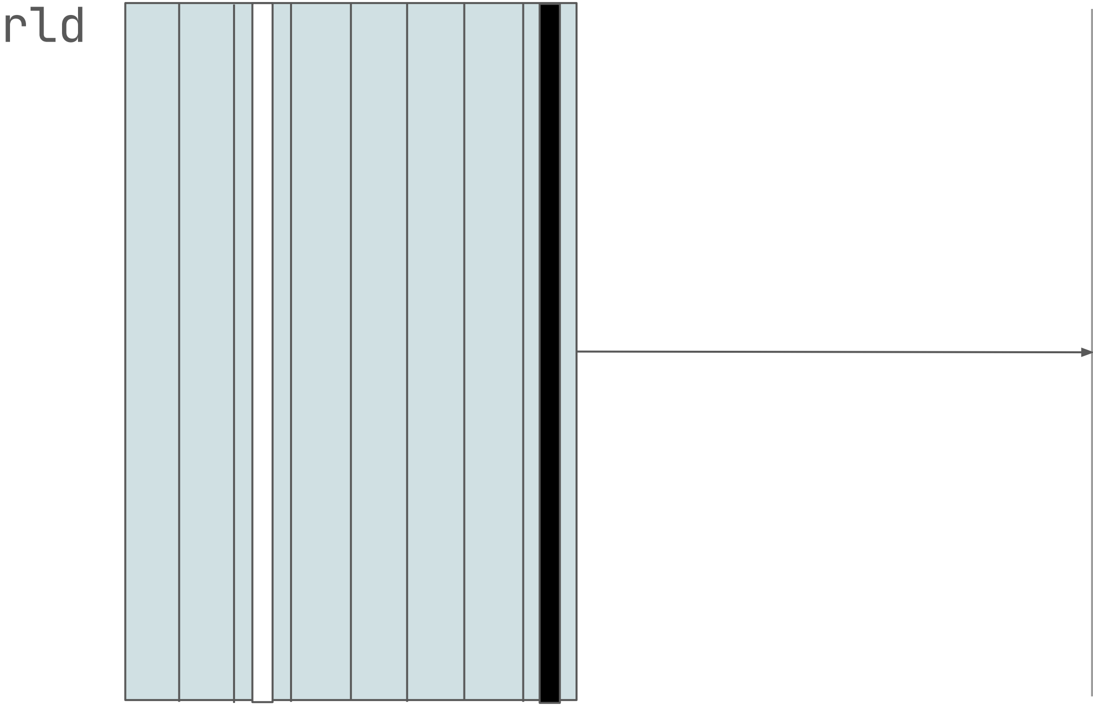
batch	tokens	dimension
1	4	8

Residual Streams After Attention and MLP



batch	tokens	dimension
1	4	8

Unembed



.	60%
!	20%
,	8%
…	…

In conclusion

- Residual stream = shared memory
- Attention: mixes between positions
- MLP: refines per position
- Embed → [Attention + MLP]^n → Unembed = the same skeleton still in use by your favorite LLM
- Any questions?

Resources / My path to this understanding

- **A Mathematical Framework for Transformer Circuits:** this paper has a section “Transformer Overview” that has influenced how I think about transformers more than anything else.
- **The papers mentioned in previous slides:** I found working through the GPT-1 paper really valuable. The *“Attention is All You Need”* paper is seminal and many people have created resources centering around it.
- **YouTube:** Serrano.Academy, Neel Nanda, Andrej Karpathy
- **Podcasts:** M*achine Learning Street Talk*, *The Cognitive Revolution*, *Latent Space*, + anything I can find with researchers from the frontier labs
- **Hands on:** I learned a lot from fine-tuning GPT-1
- **Claude and ChatGPT**