Combining LLMs and Cognitive Models of Memory

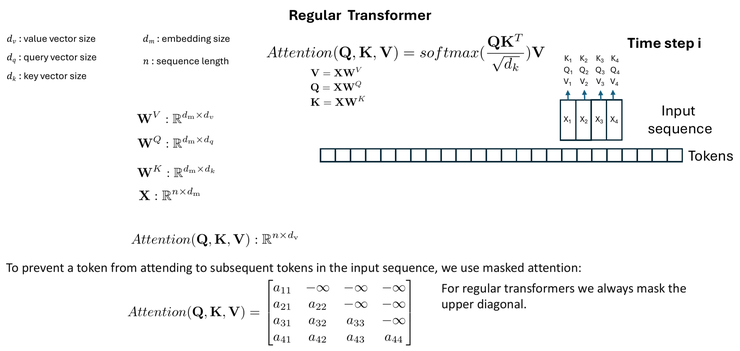
Billy Dickson [1] Sahaj Singh Maini [1] Zoran Tiganj [1]

[1]Indiana University

Regular Transformer

Attention Mechanism: The core of the transformer is the attention mechanism, which allows each token to attend to every other token in the fixed window size of the sequence. This is achieved through:

- **Query, Key, and Value Vectors:** Each token is represented by three vectors. The query vector of each token is compared to the key vectors of all other tokens to establish attention scores.
- **Softmax Normalization:** Attention scores are normalized using the softmax function, making them sum to one. This process allows the model to decide the relevance or importance of other tokens relative to each token.
- **Weighted Sum:** The value vectors are then weighted by the normalized attention scores and summed to produce the output of the attention layer for each token.



To prevent a token from attending to subsequent tokens in the input sequence, we use masked attention:

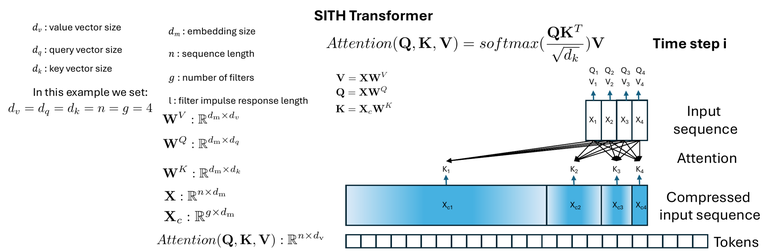
$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = \begin{bmatrix} a_{11} & -\infty & -\infty & -\infty \\ a_{21} & a_{22} & -\infty & -\infty \\ a_{31} & a_{32} & a_{33} & -\infty \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

For regular transformers we always mask the upper diagonal.

Figure 1. Regular Transformer Attention Mechanism

SITH (Scale Invariant Temporal History) Transformer

- Log Compressed Key Attention Mechanism:
  - **Memory Compression:** A longer sequence of past inputs are compressed and used as keys.
  - **Query Generation:** Queries are created from current token embeddings.
  - **Attention Scores:** Computed using dot products between queries and compressed keys, followed by softmax normalization.
  - **Value Vector:** Vale vector is weighted by the normalized attention scores derived from this compressed history.



To prevent a token from attending to subsequent tokens in the input sequence, we use masked attention:

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = \begin{bmatrix} a_{11} & -\infty & -\infty & -\infty \\ a_{21} & a_{22} & -\infty & -\infty \\ a_{31} & a_{32} & a_{33} & -\infty \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = \begin{bmatrix} a_{11} & \mathbf{a_{12}} & -\infty & -\infty \\ a_{21} & a_{22} & -\infty & -\infty \\ a_{31} & a_{32} & a_{33} & -\infty \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

For SITH attention, we threshold the filters and then check which of the filters contain the future information relative to the current token and mask accordingly: for example, note that a_{12} is not masked in SITH attention because filter X_{c2} is the one emitting the key and that filter contains only information before word X_1 (which emits query and value).

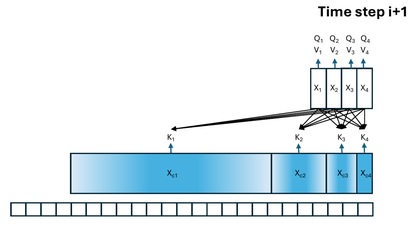


Figure 2. SITH Transformer Attention Mechanism

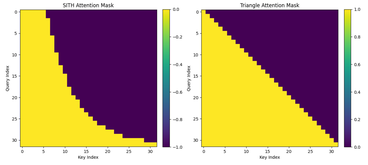


Figure 3. SITH Attention Mask vs. Regular Attention Mask

Log Compressed Filter Convolution

This set of log compressed filters are convolved over the input each representing a different resolution.

- The **more recent past** is higher resolution but contains less words.
- The **more distant past** is lower resolution and contains a compression of more words.
- Each filter is one key in the attention mechanism.

- Regular Transformer asks "**how much of this other word is helpful in predicting the next token**"
- SITH transformer asks "**how much of this compressed representation of many words is useful in predicting the next token**".

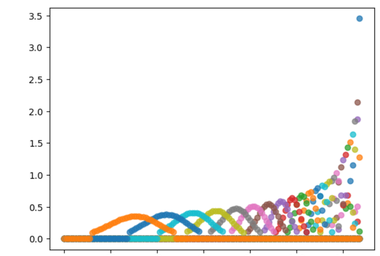


Figure 4. Log Spaced Compression Filters

A set of delta pulse filters are used as control for comparison. These don't compress the input, and simply pass the information through as if it were normal attention.

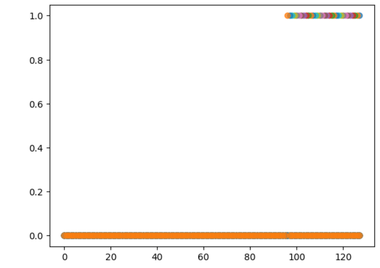


Figure 5. Delta Pulse Filters (used as control)

Performance on Wikitext-103

Model	Param	Seq Length	PPL
SITH (delta pulses)	135M	64	30.56
SITH	135M	64 (256)	25.40

Table 1. Performance on Wikitext-103

{dicksonb} {sahmaini} {ztiganj} @iu.edu