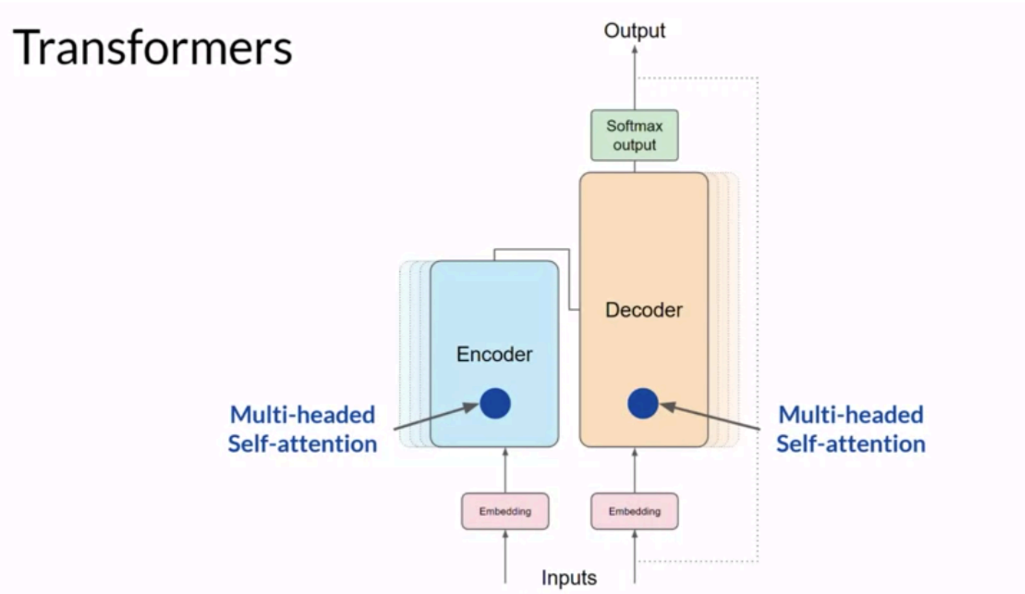
Understanding Large Language Models with Transformer Architecture

- ## Introduction:
  - Transformer architecture drastically improved natural language task performance.
  - Shifting from RNNs to transformers led to enhanced regenerative capability.
  - Key feature: learning word relevance and context through attention.

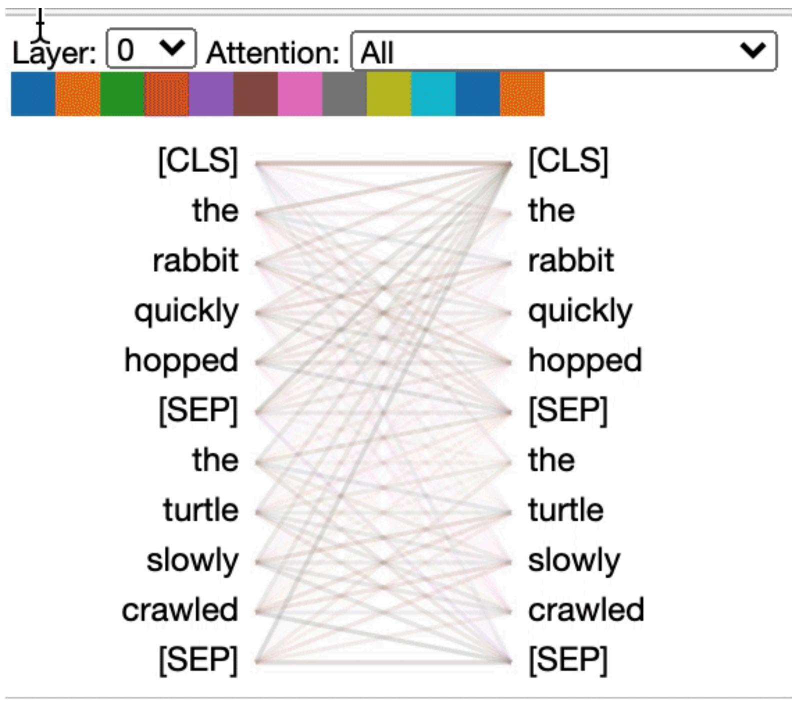


- ## Transformer Architecture Overview:
  - Divided into encoder and decoder parts, working together.
  - Derived from the "Attention is All You Need" paper.
  - Inputs at bottom, outputs at top for consistency.

- # Tokenization and Embedding:
  - Words tokenized into numerical representations.
  - Tokenization is crucial for model training and generation.
  - Embedding layer maps tokens to high-dimensional vector space.
  - Each token has a unique vector, encoding meaning and context.

- # Positional Encoding:
  - Maintains word order relevance via positional encodings.
  - Added to token vectors to preserve sentence structure.

- # Self-Attention:
  - Model analyzes token relationships using self-attention.
  - Self-attention applied across the input sequence.
  - Multi-headed self-attention enables learning diverse language aspects.
  - Each attention head focuses on different linguistic properties.



- # Output Processing:
  - Output processed through a fully-connected feed-forward network.

- Resulting logits represent token probability scores.
- Softmax layer normalizes logits into probability distributions.
- Final token selection based on highest probability.

Final Note

- Transformer architecture enables robust language understanding.
- Continuous training enhances model capabilities over time.

REFERENCES

- https://jalammar.github.io/illustrated-transformer/