Introduction to Transformer Architecture

Presented by:
Abhisek Keshari

Research Supervisor:
Dr. Virendra Singh

Outline

- Introduction : Attention is All You Need Transformer
- Workflow Diagram
- Input Methods - Tokenizing the input data
- Embeddings
- Positional Encoding
- Encoder Block

Outline

- Self Attention Unit
- Why Multi-Head Attention
- Decoder Block
- Look-Ahead Mask
- Output Probabilities

Introduction

Attention is All You Need Transformer :

- **No Sequential Processing :** Dispenses with sequential processing, employing self-attention to establish global dependencies.
- **Parallel Computations :** Allows parallelization of computations, enhancing efficiency and scalability.
- **Enhanced Performance :** Revolutionizes tasks like language translation and image analysis, outperforming traditional sequential models.

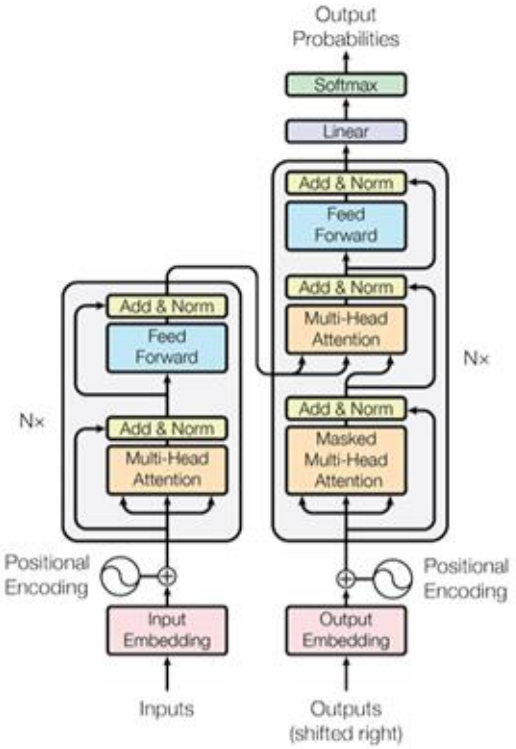
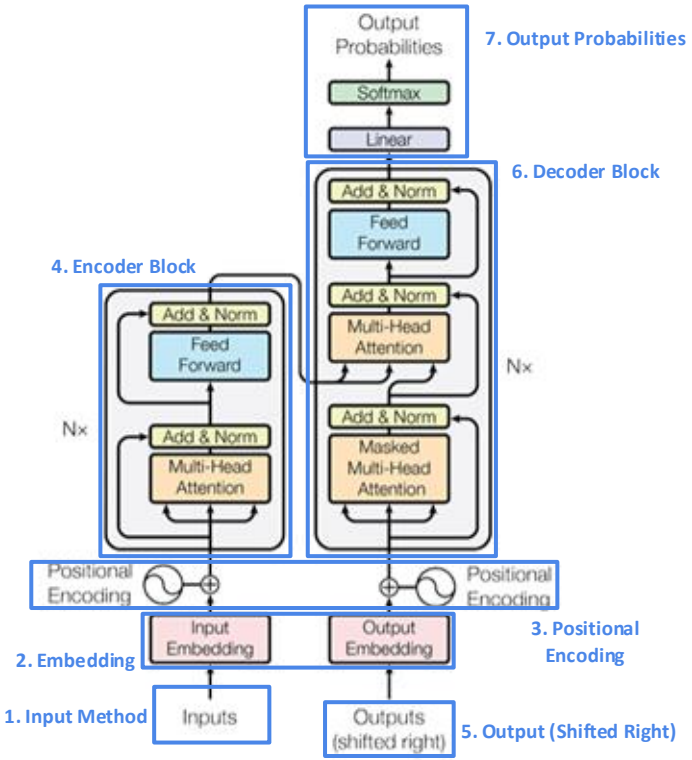


Figure 1: The Transformer - model architecture.

Workflow Diagram

1. Input Methods
2. Embedding
3. Positional Encoding
4. Encoder Block
5. Output (Shifted Right)
6. Decoder Block
7. Output Probabilities

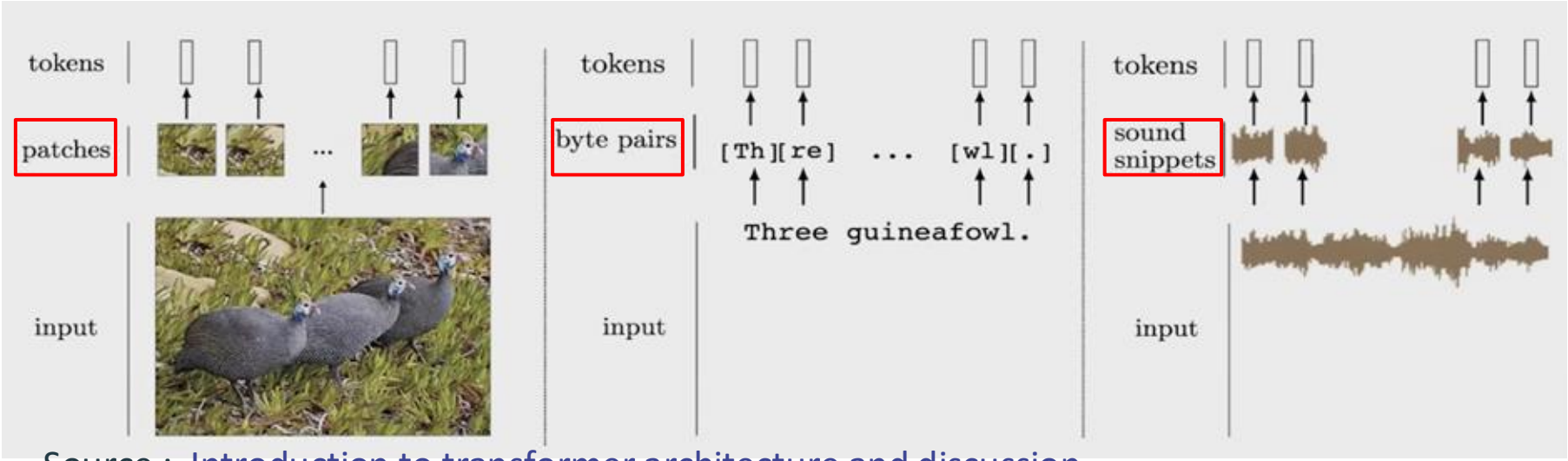


The Transformer - model architecture.

Input Methods - Tokenizing the input data

General Strategy :

Chop the input up into chunks, and project each chunk into vector



Source : [Introduction to transformer architecture and discussion](), https://vaclavkosar.com/ml/Tokenization-in-Machine-Learning-Explained

Embedding

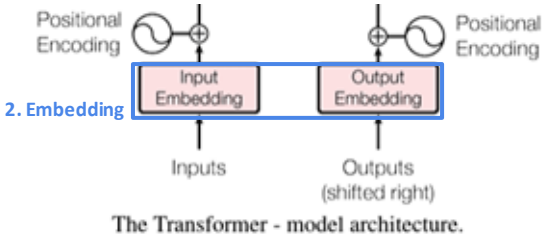
First we feed input into a embedding layer.

A **word / sequence** layer can be thought of as a lookup table to grab a learned vector representation of **word / sequence**.

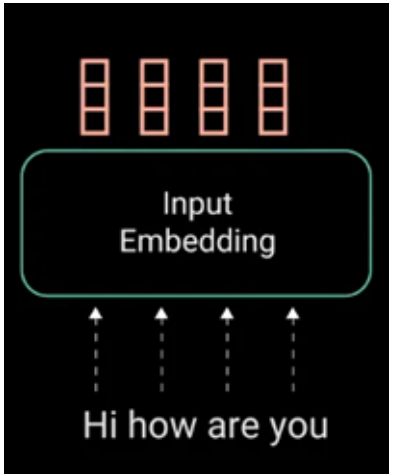
Neural networks learn through numbers so each word maps to a vector with continuous values to represent that word / sequence.

Popular word embedding methods:

- Word2Vec
- TF-IDF



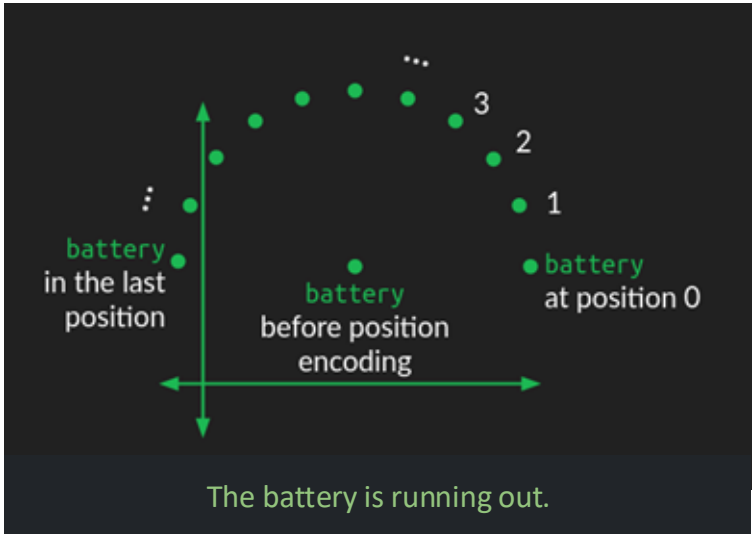
The Transformer - model architecture.



Ref: https://vaclavkosar.com/ml/Embeddings-in-Machine-Learning-Explained

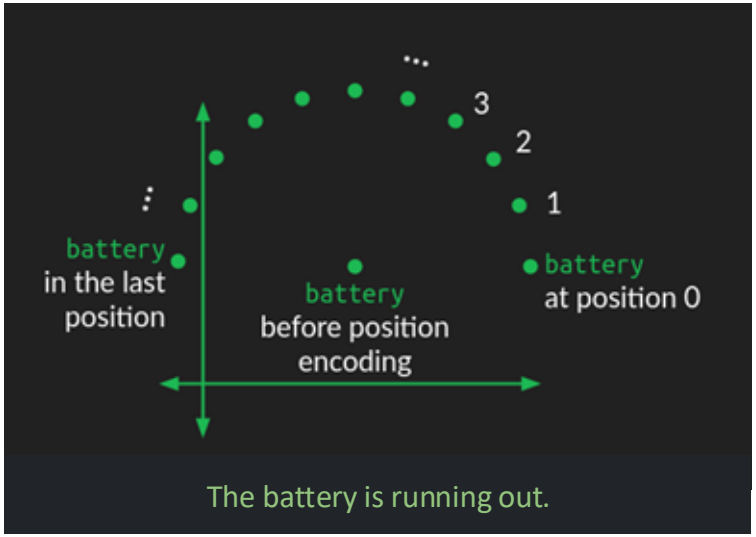
Positional Encoding (1/5)

- **Embedding Space as Center:** Word positions in the embedding space act as centers of circles.
- **Position-dependent Perturbation:** A perturbation is added based on the word's position in the sequence, creating a consistent circular pattern.
- **Angular Variation:** Words are perturbed at different angles but with the same distance, forming a circular pattern in the sequence.



Positional Encoding (2/5)

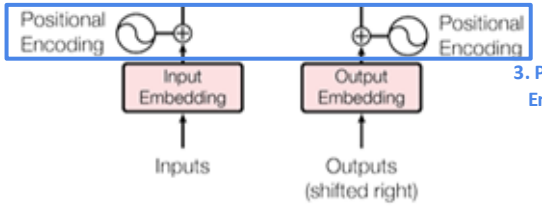
- **Proximity Influence:** Close sequence words experience similar perturbations, while distant ones undergo variations in different directions.



Positional Encoding (3/5)

A positional encoding is a finite dimensional representation of the location or “position” of items in a sequence.

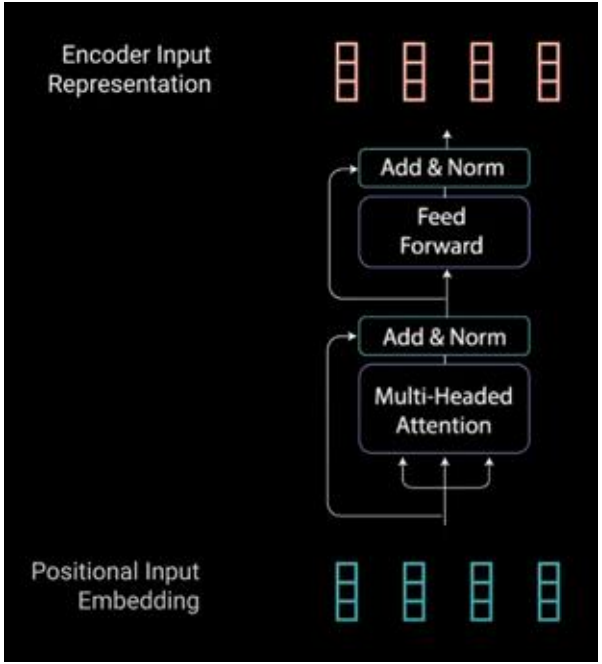
Given some sequence **A = [a_0, …, a_{n-1}]**, the positional encoding must be some type of tensor that we can feed to a model to tell it where some value **a_i** is in the sequence A.



The Transformer - model architecture.

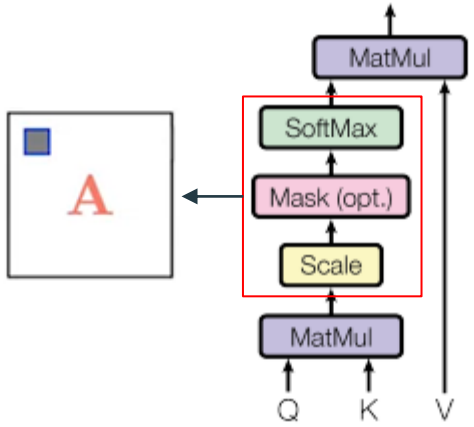
Encoder Block

- Encoder map all input sequences into an abstract continuous representation that holds the learned information for that entire sequence.
- It contains 2 sub-modules, **multi-headed attention**, followed by a **fully connected network**.
- There are also **residual connections** around each of the two sublayers followed by a layer **normalization**.



Self Attention Unit

- The main purpose of self attention mechanism is to add contextual information to words in sentence.



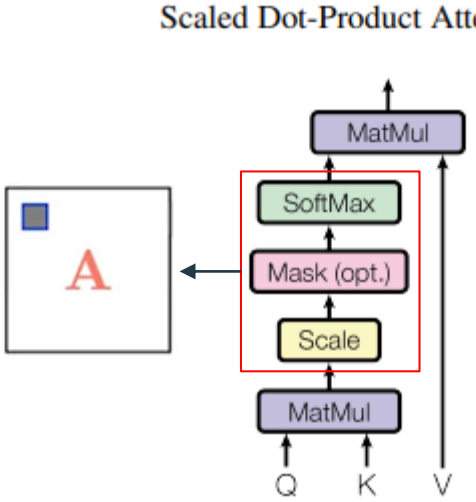
…government debt problems turning into ***banking*** *crises as happened in 2009…*
…saying that Europe needs unified ***banking*** *regulation to replace the hodgepodge…*
…India has just given its ***banking*** *system a shot in the arm…*

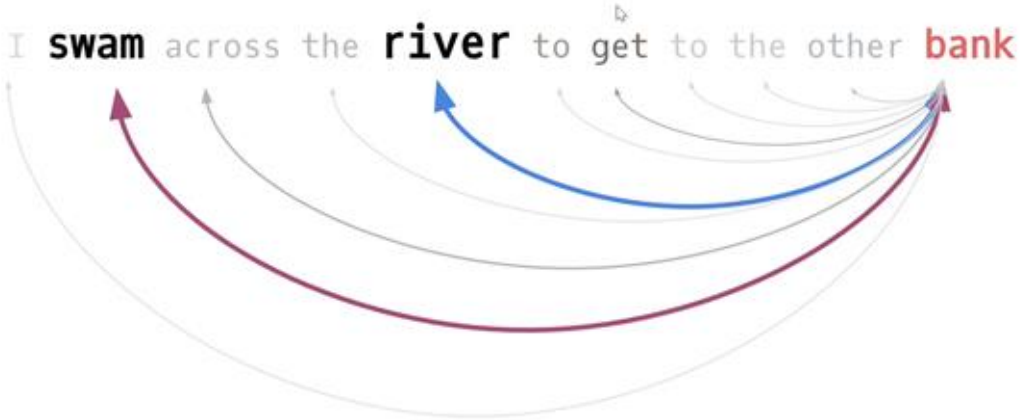
These context words will represent ***banking***

Source : CS224N - Lecture 1

Self Attention Unit

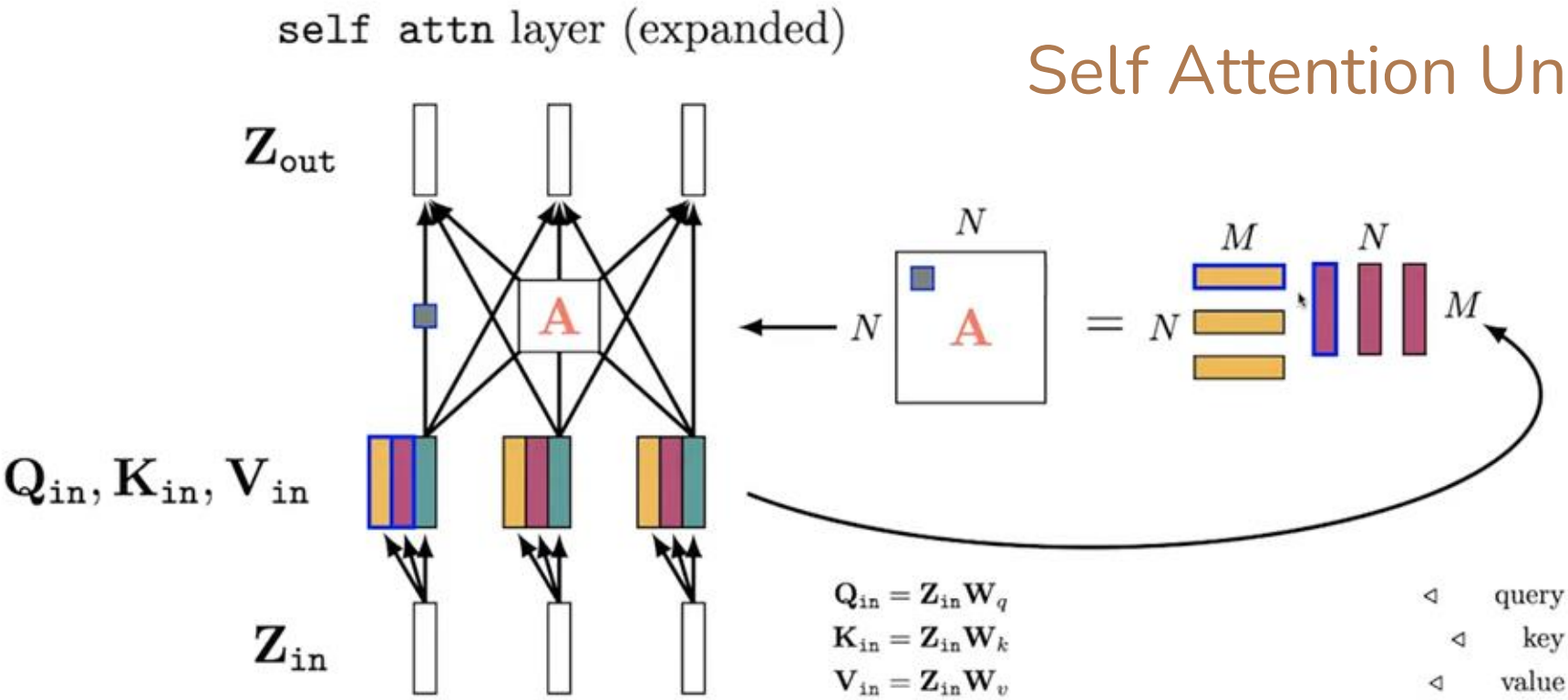
- Q : The information that we want to query on
- K : The information that we already have
- V : What value we are adding after attention unit.





[Over to the board]

Self Attention Unit



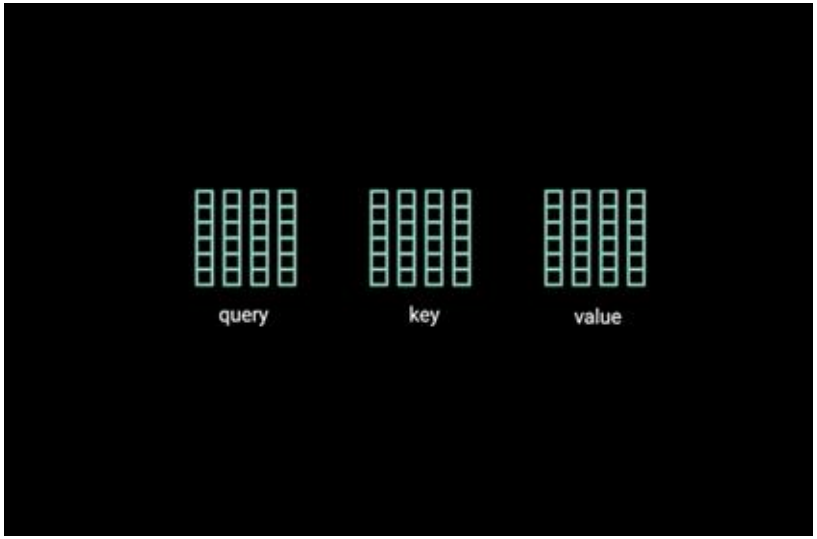
$$\mathbf{Q}_{\text{in}} = \mathbf{Z}_{\text{in}}\mathbf{W}_q \quad \triangleleft \text{ query matrix}$$
$$\mathbf{K}_{\text{in}} = \mathbf{Z}_{\text{in}}\mathbf{W}_k \quad \triangleleft \text{ key matrix}$$
$$\mathbf{V}_{\text{in}} = \mathbf{Z}_{\text{in}}\mathbf{W}_v \quad \triangleleft \text{ value matrix}$$

$$\mathbf{A} = f(\mathbf{t}_{\text{in}}) = \texttt{softmax}(\frac{\mathbf{Q}_{\text{in}}\mathbf{K}_{\text{in}}^T}{\sqrt{d}}) \quad \triangleleft \text{ attention matrix}$$

$$\mathbf{Z}_{\text{out}} = \mathbf{A}\mathbf{V}_{\text{in}}$$

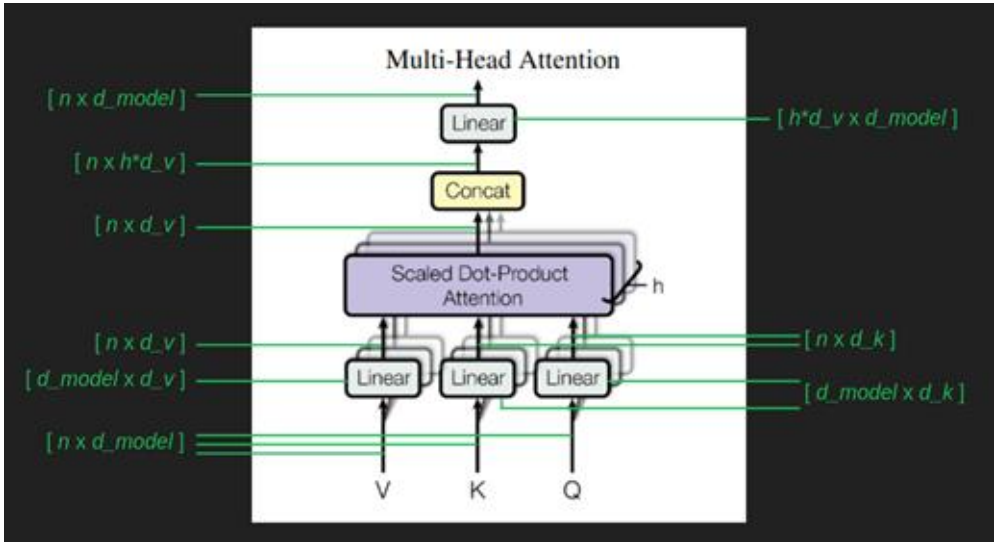
Multi-Head Attention

- To make this a multi-headed attention computation, you need to split the query, key, and value into N vectors before applying self-attention.
- The split vectors then go through the self-attention process individually.
- Results of these heads get concatenated.

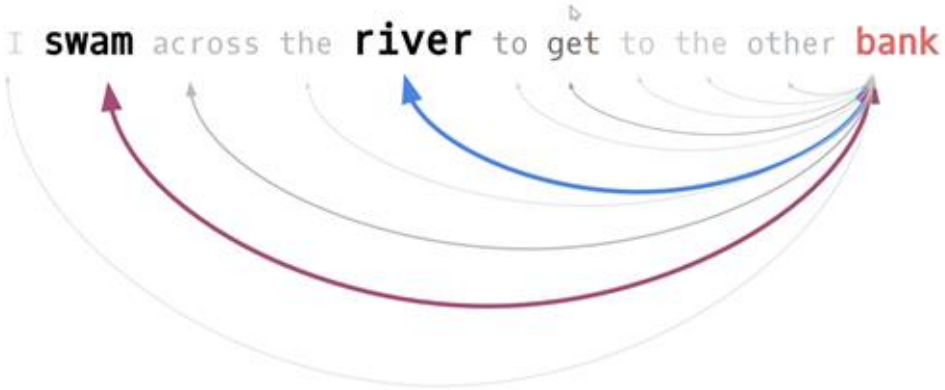


Why Multi-Head Attention (1/2)

- In a single samples there could be multiple contexts.
- To capture these contexts we need to have Multi-Head Attention.

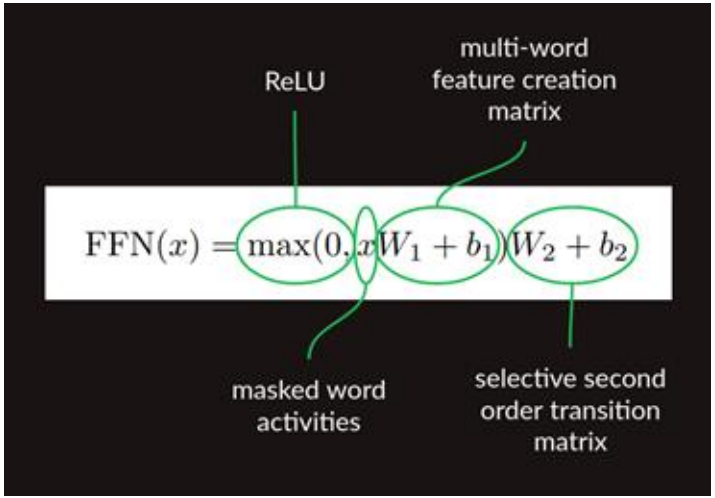


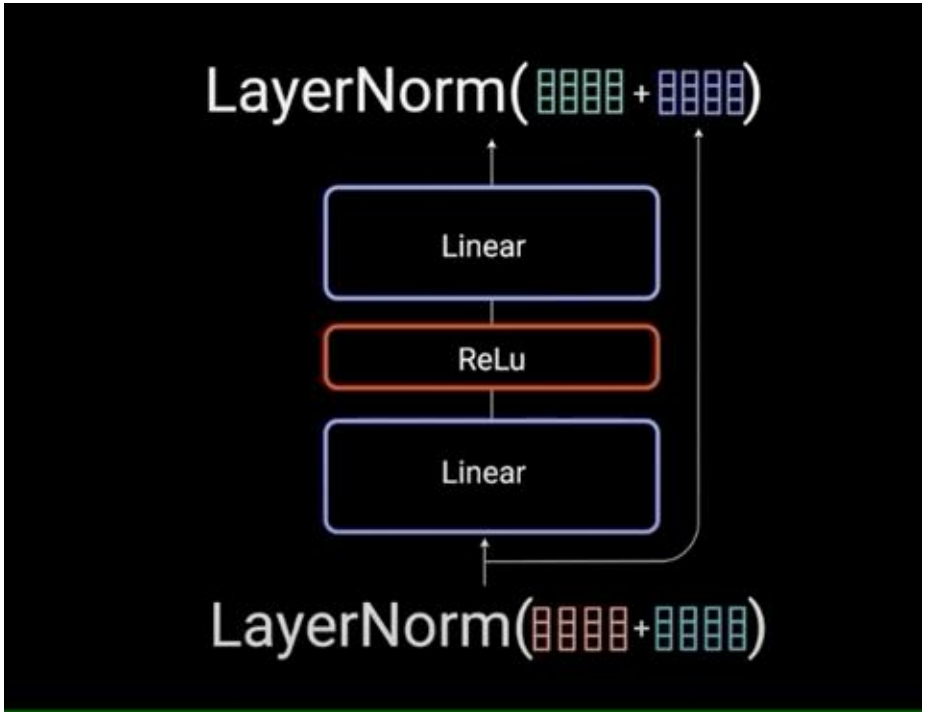
Why Multi-Head Attention (2/2)



Significance of ReLu

To normalize the score from attention unit, keep the pair with the maximum resemblance to the context.



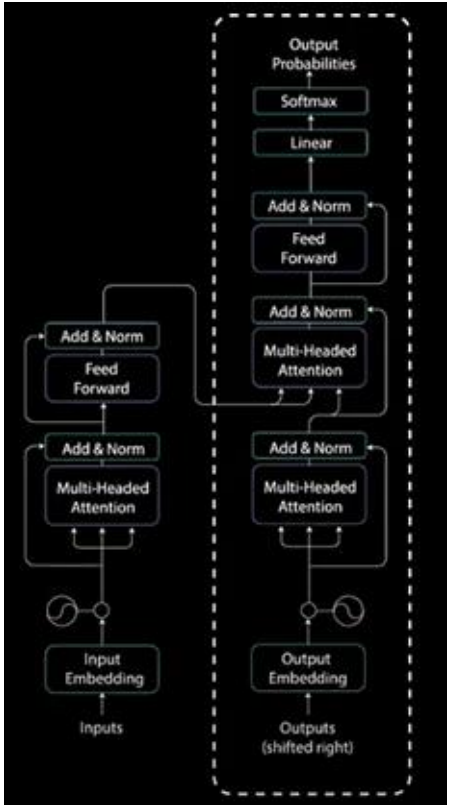


Decoder Block (1/2)

This block has a similar sub-layer as the encoder. It has two **multi-headed attention layers**, a pointwise **feed-forward layer,** and r**esidual connections**, and **layer normalization** after each sub-layer.

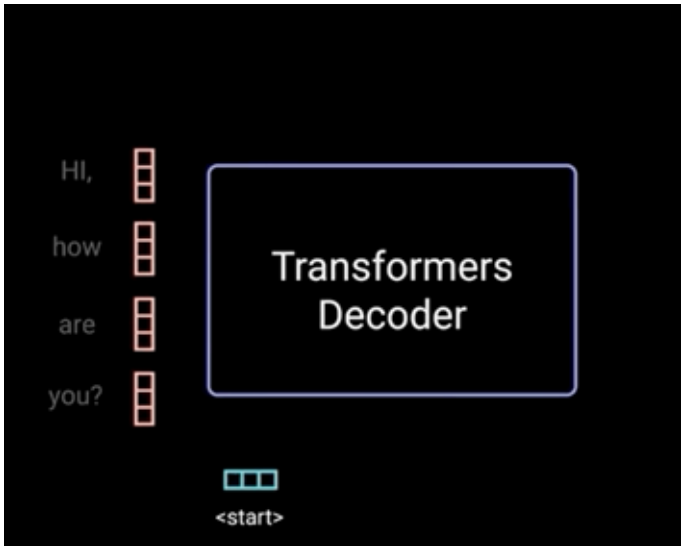
These sub-layers behave similarly to the layers in the encoder but each multi-headed attention layer has a different job.

The decoder is capped off with a linear layer that acts as a classifier, and a softmax to get the word probabilities.



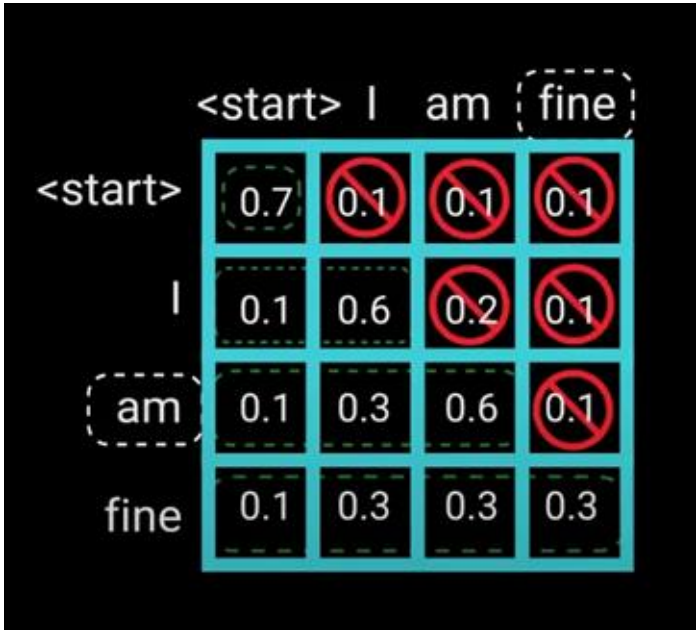
Decoder Block (2/2)

- The decoder is autoregressive, it begins with a start token, and it takes in a list of previous outputs as inputs, as well as the encoder outputs that contain the attention information from the input.
- The decoder stops decoding when it generates a token as an output.
- The decoder's job is to generate text sequences.



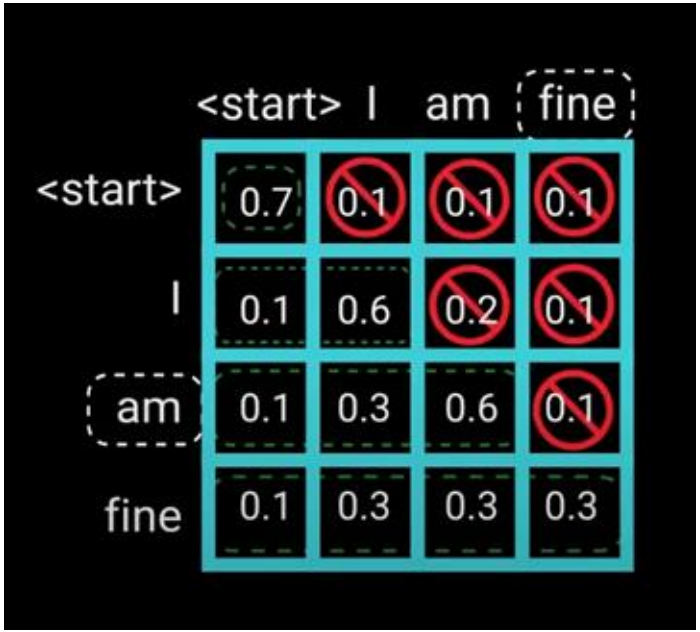
Decoder Block (2/2)

- Since the decoder is autoregressive and generates the sequence word by word, We need to prevent it from conditioning to future tokens.



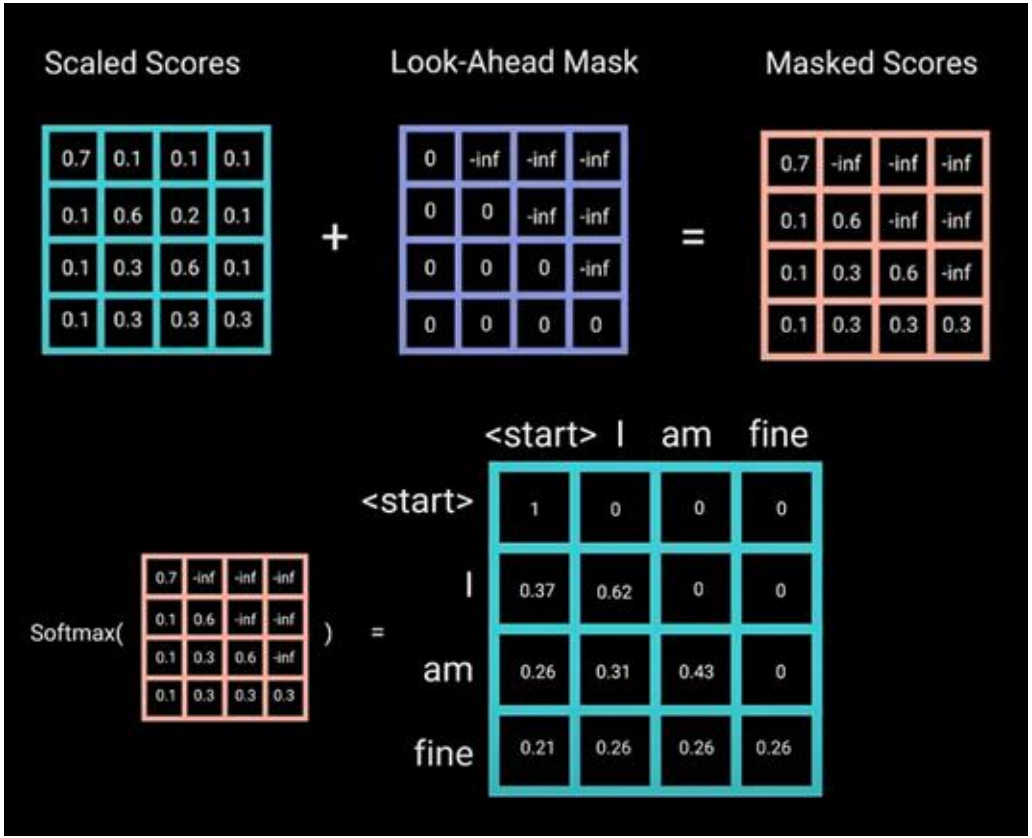
Look-Ahead Mask (1/2)

- The attention mechanism employs a matrix mask, matching the size of attention scores.
- Filled with 0's and negative infinities, it highlights the top right triangle of the scaled attention scores matrix.
- Upon adding the mask to scaled attention scores, the top right triangle is populated with negative infinities.
- After applying softmax, these infinities become zeros, essentially nullifying attention to future tokens.



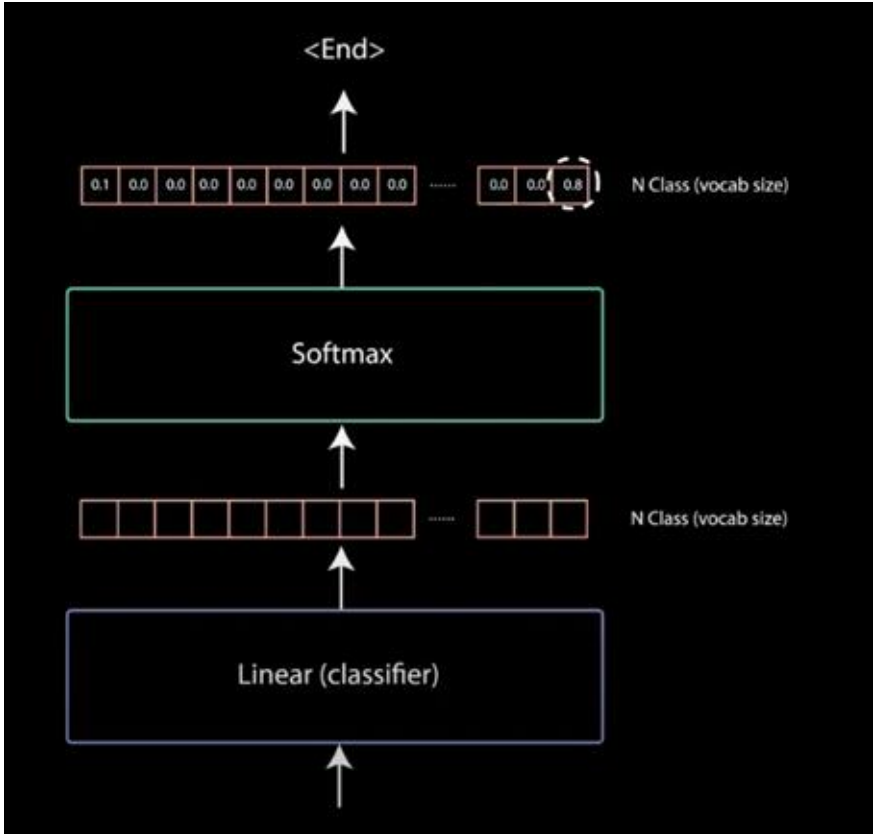
Look-Ahead Mask (2/2)

This masking is the only difference in how the attention scores are calculated in the first multi-headed attention layer.



Output Probabilities

The output of the final pointwise feedforward layer goes through a final linear layer, that acts as a classifier. The classifier is as big as the number of classes you have.



Transformer Arch.

Further Reading

- Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017). (**Main**)
- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems* 27 (2014). (**Seq-to-Seq**)
- Illustrated Guide to Transformers- Step by Step Explanation
- The Illustrated Transformer
- Transformers from Scratch
- Understanding LSTM Networks
- Word Vectors

Transformer Arch.

Further Reading

- Wang, Changhan, Kyunghyun Cho, and Jiatao Gu. "Neural machine translation with byte-level subwords." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 05. 2020.

References

- Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017). (**Main**)
- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems* 27 (2014). (**Seq-to-Seq**)
- Illustrated Guide to Transformers- Step by Step Explanation
- The Illustrated Transformer
- Transformers from Scratch
- Understanding LSTM Networks
- Word Vectors
- Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units." *arXiv preprint arXiv:1508.07909* (2015).

Questions?

Thank You