POST 21 - GEN AI

BEYOND THE SURFACE: UNDERSTANDING HOW ENCODERS WORK IN TRANSFORMERS

Generative AI Deep Dives,
Key concepts for Transformers - Part 6

GENERATIVE AI FOR ALL

DINESH LAL
(DIRECTOR, DATA SCIENCE)

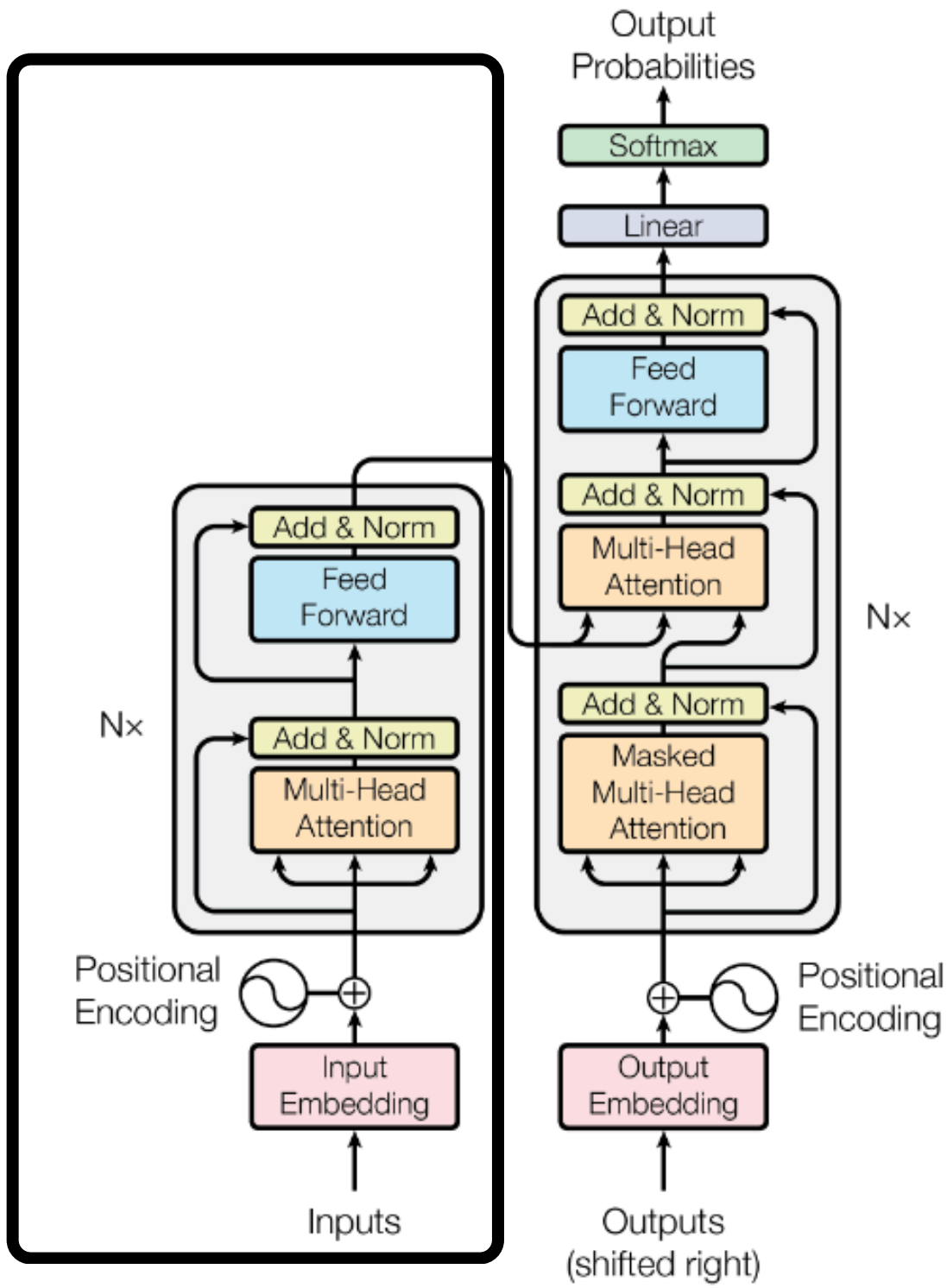
DINESHLAL

WHAT IS COVERED IN THIS DOCUMENT?

- This document explains the topic **Encoder** in Transformer Architectue
- First visual representation is covered
- Then Definition of Encoder, and simple explanation is shared
- In the detailed section, step by step explanation is covered with
  - Each stage explanation
  - Each stage inputs and Ouputs,
  - An Example to explain the concepts better

VISUAL REPRESENTATION OF ENCODER

ENCODER →



DEFINING ENCODER

- The encoder in a Transformer architecture plays a crucial role in **processing input data** and preparing it for further processing by the decoder. It consists of **several layers**,
- Each comprising two main components: **self-attention mechanism** and **feedforward neural network**. The encoder operates sequentially, with each layer transforming the input data through a series of operations
- Imagine a conductor in a giant orchestra. **Their job is to listen intently to all the musicians and understand how each instrument contributes to the overall sound.** This is exactly what the encoder in a transformer architecture does! Instead of musical notes, the encoder works with words or other elements in a sequence of data.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

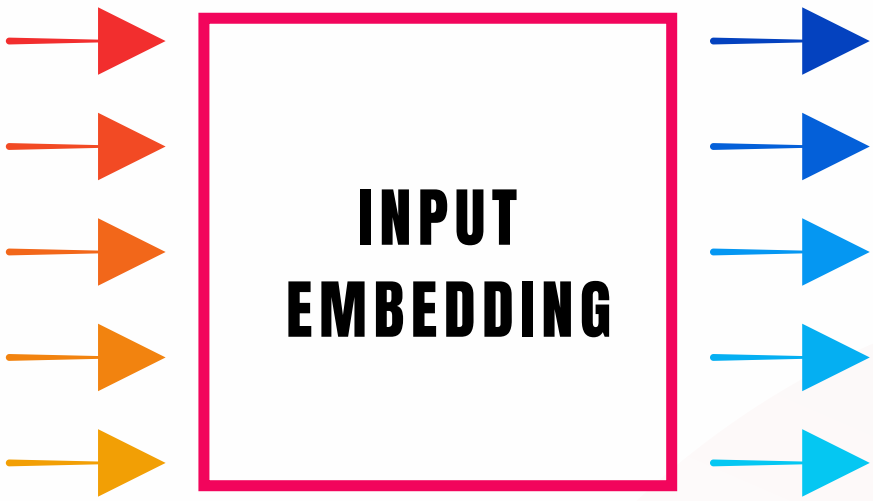
1 INPUT EMBEDDING

- The input sequence, typically represented as a sequence of word embeddings or token embeddings, is fed into the encoder.
- Each token in the input sequence is transformed into a high-dimensional embedding vector that represents its semantic meaning in the context of the sequence.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

1 INPUT EMBEDDING

Inputs:
Sequence of token embeddings representing the input text.

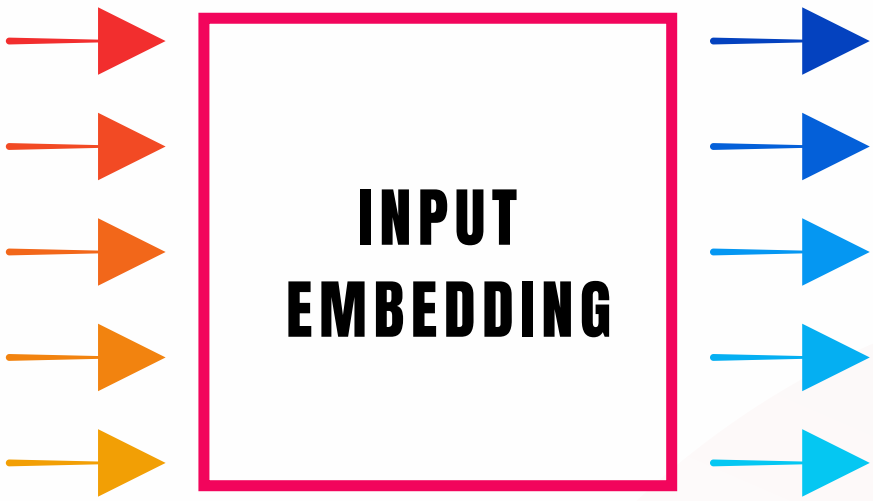


Outputs
Embedded representations of each token in the input sequence.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

1 INPUT EMBEDDING

Inputs:
The Cat sat on the Mat.



Outputs
[embedding("The"),
embedding("cat"),
embedding("sat"),
embedding("on"),
embedding("the"),
embedding("mat")]

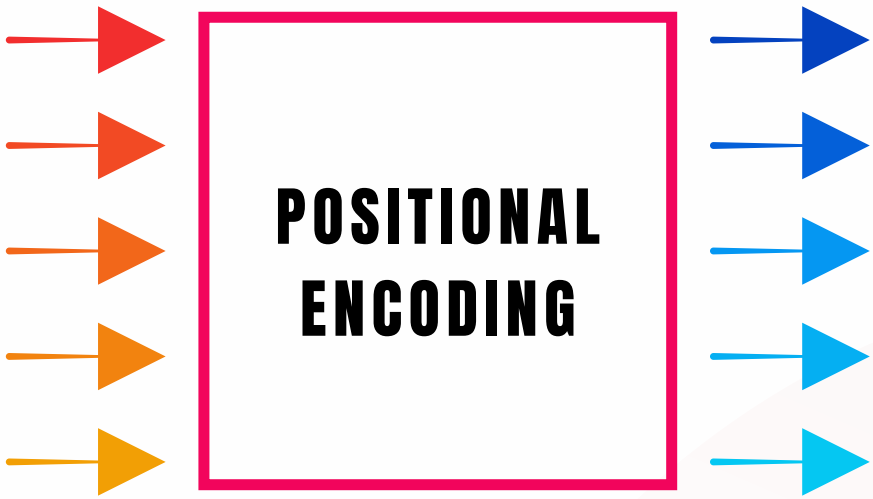
2 POSITIONAL ENCODING

- Since Transformers do not inherently understand the order of tokens in a sequence, positional encoding is added to provide information about the position of tokens.
- Positional encoding vectors are added to the input embeddings, allowing the model to differentiate between tokens based on their position in the sequence.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

2 POSITIONAL ENCODING

Inputs:
Embedded representations of tokens.

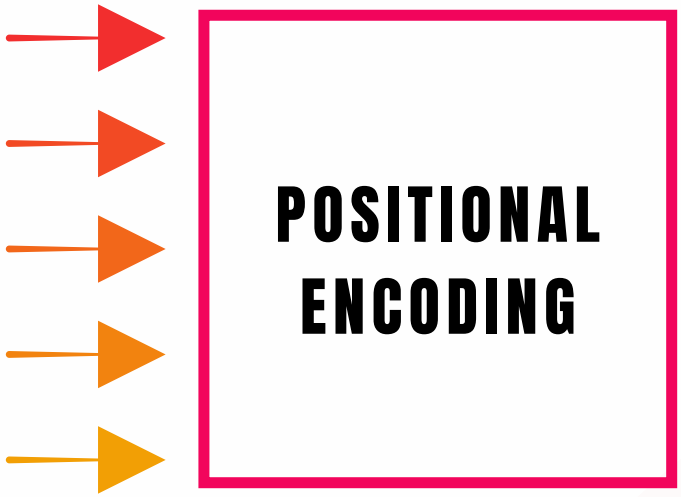


Outputs
Token embeddings with positional encoding added, preserving both token semantics and positional information.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

2 POSITIONAL ENCODING

Inputs:
[embedding("The"),
embedding("cat"),
embedding("sat"),
embedding("on"),
embedding("the"),
embedding("mat")]



Outputs
[embedding("The") +
positional_encoding(1),
embedding("cat") +
positional_encoding(2),
embedding("sat") +
positional_encoding(3),
embedding("on") +
positional_encoding(4),
embedding("the") +
positional_encoding(5),
embedding("mat") +
positional_encoding(6)]

3 SELF-ATTENTION MECHANISM

- The core of the encoder is the self-attention mechanism, which enables the model to weigh the importance of different tokens in the input sequence when processing each token.
- Self-attention computes attention scores between all pairs of tokens in the input sequence and generates context-aware representations for each token.
- It allows the model to focus more on relevant tokens and less on irrelevant ones, capturing long-range dependencies effectively.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

3

SELF-ATTENTION MECHANISM

Inputs:
Token embeddings with positional encoding..

Outputs

- Contextualized representations of tokens obtained through self-attention mechanism.
- Each token representation captures its relationship with other tokens in the sequence.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

3 SELF-ATTENTION MECHANISM

Inputs:

[embedding("The") + positional_encoding(1),
embedding("cat") + positional_encoding(2),
embedding("sat") + positional_encoding(3),
embedding("on") + positional_encoding(4),
embedding("the") + positional_encoding(5),
embedding("mat") + positional_encoding(6)]

Outputs:

[contextualized_embedding("The"),
contextualized_embedding("cat"),
contextualized_embedding("sat"),
contextualized_embedding("on"),
contextualized_embedding("the"),
contextualized_embedding("mat")]

4

MULTI-HEAD ATTENTION

- To capture different aspects of the input sequence, self-attention is often performed multiple times in parallel, each with different learned projection matrices.
- These parallel self-attention mechanisms are called "attention heads," and they allow the model to attend to different parts of the input sequence simultaneously.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

4

MULTI-HEAD ATTENTION

Inputs:
Contextualized representations of tokens from self-attention mechanism.

Outputs
Enhanced representations of tokens with multiple perspectives, obtained through parallel self-attention heads.

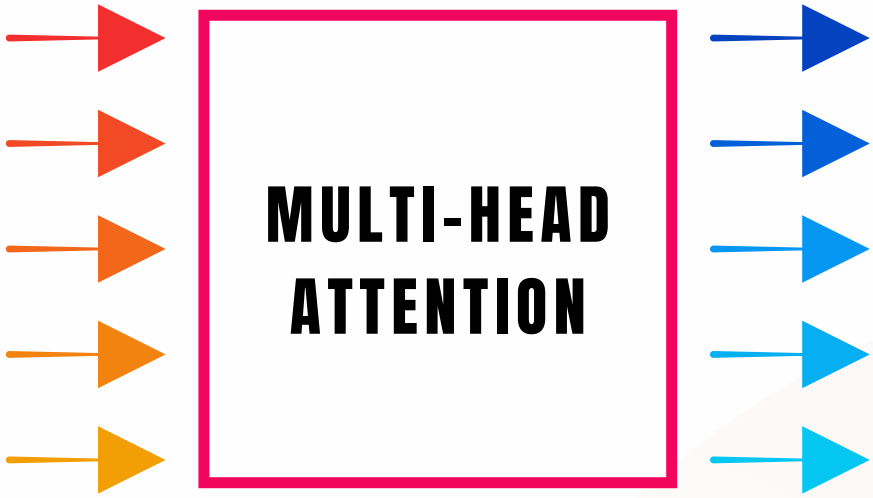
STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

4

MULTI-HEAD ATTENTION

Inputs:

[contextualized_embedding("The"), contextualized_embedding("cat"), contextualized_embedding("sat"), contextualized_embedding("on"), contextualized_embedding("the"), contextualized_embedding("mat")]



Outputs:

[[enhanced_contextualized_embedding("The"), enhanced_contextualized_embedding("cat"), enhanced_contextualized_embedding("sat"), enhanced_contextualized_embedding("on"), enhanced_contextualized_embedding("the"), enhanced_contextualized_embedding("mat")]

DINESHLAL

5 FEEDFORWARD NEURAL NETWORK (FFN):

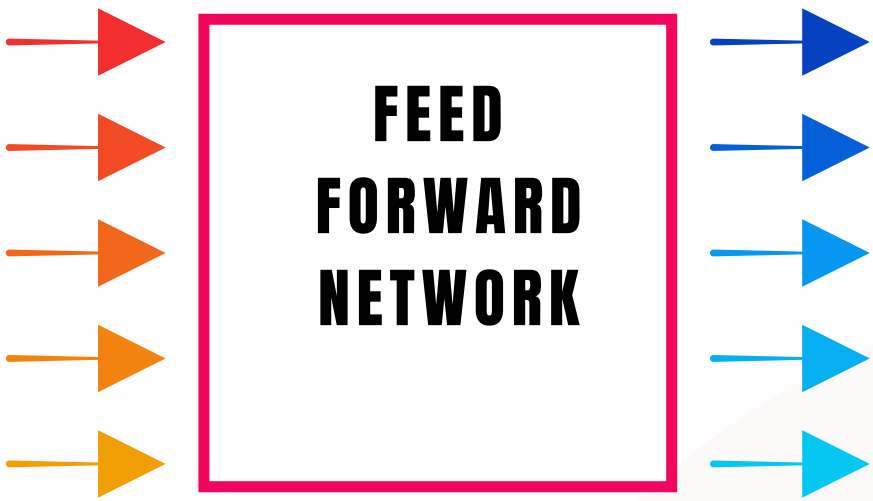
- After self-attention, the output from each attention head is concatenated and passed through a feedforward neural network.
- The FFN consists of two linear transformations separated by a non-linear activation function, such as ReLU.
- It enables the model to capture complex patterns and relationships within the input sequence.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

5

FEEDFORWARD NEURAL NETWORK (FFN):

Inputs:
Contextualized representations of tokens, potentially from multiple attention heads.



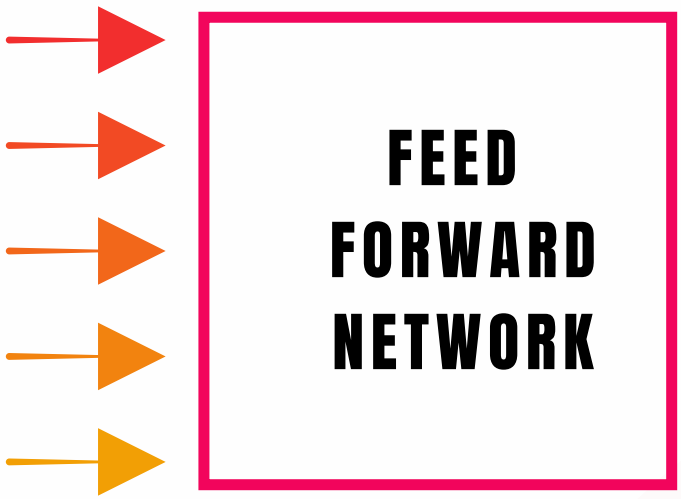
Outputs:
Transformed representations of tokens after passing through the feedforward neural network. This captures complex patterns and relationships within the input sequence.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

5 FEEDFORWARD NEURAL NETWORK (FFN):

Inputs:

[contextualized_embedding("The"), contextualized_embedding("cat"), contextualized_embedding("sat"), contextualized_embedding("on"), contextualized_embedding("the"), contextualized_embedding("mat")]



Outputs:

[transformed_embedding("The"), transformed_embedding("cat"), transformed_embedding("sat"), transformed_embedding("on"), transformed_embedding("the"), transformed_embedding("mat")]

6

LAYER NORMALIZATION AND RESIDUAL CONNECTION:

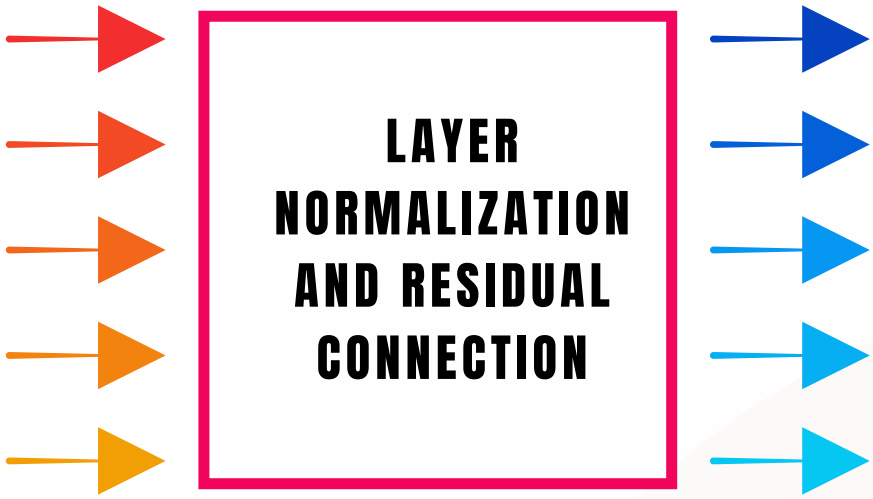
- To stabilize the training process and facilitate the flow of gradients, layer normalization is applied after each sub-layer (self-attention and feedforward network).
- Additionally, residual connections are employed, allowing the original input to bypass the sub-layers and be summed with the output.
- This helps alleviate the vanishing gradient problem and facilitates training deeper networks.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

6

LAYER NORMALIZATION AND RESIDUAL CONNECTION:

Inputs:
Transformed representations of tokens from the feedforward neural network.



Outputs:
Normalized representations with residual connections applied, preserving information from previous layers while stabilizing training.

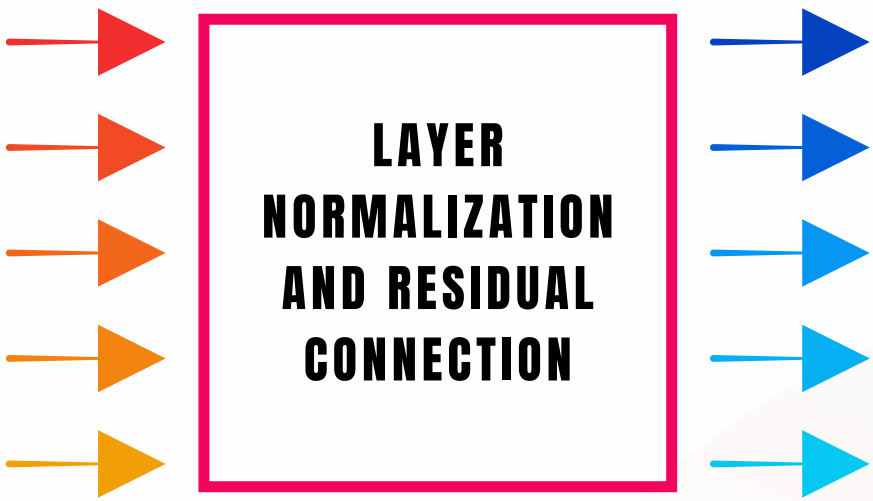
STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

6

LAYER NORMALIZATION AND RESIDUAL CONNECTION:

Inputs:

[transformed_embedding("The"),
transformed_embedding("cat"),
transformed_embedding("sat"),
transformed_embedding("on"),
transformed_embedding("the"),
transformed_embedding("mat")]



Outputs:

[[normalized_embedding("The"),
normalized_embedding("cat"),
normalized_embedding("sat"),
normalized_embedding("on"),
normalized_embedding("the"),
normalized_embedding("mat")]

7

STACKING ENCODER LAYERS:

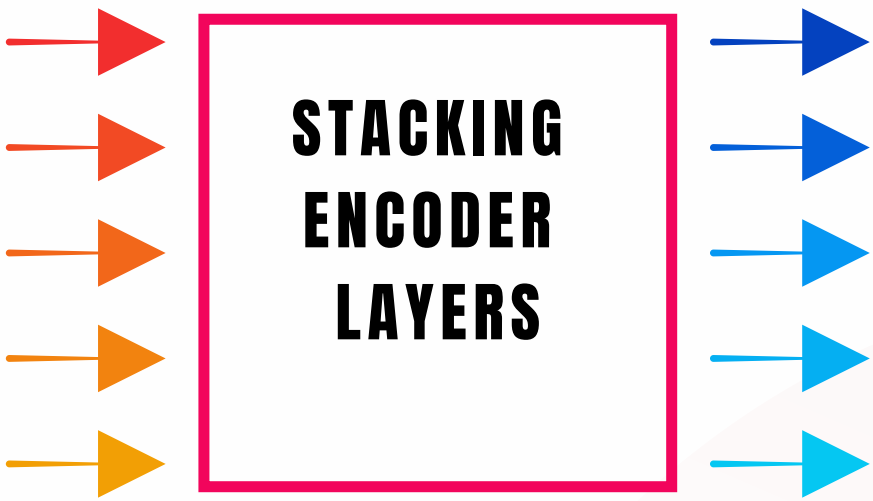
- The encoder consists of multiple identical layers, each containing a self-attention mechanism and feedforward neural network.
- The output of one encoder layer serves as the input to the next layer, allowing the model to capture increasingly complex patterns and dependencies in the input sequence.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

7

STACKING ENCODER LAYERS:

Inputs:
Output representations from the previous encoder layer



Outputs:
Contextualized representations of tokens from the current encoder layer, ready to be passed to the next layer.

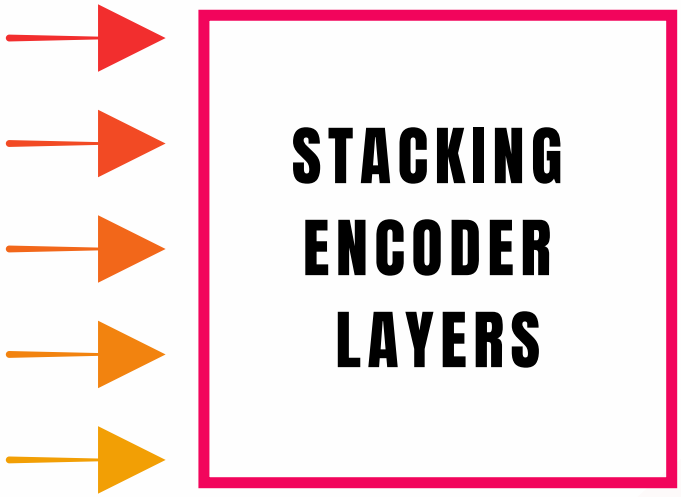
STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

7

STACKING ENCODER LAYERS:

Inputs:

[normalized_embedding("The"), normalized_embedding("cat"), normalized_embedding("sat"), normalized_embedding("on"), normalized_embedding("the"), normalized_embedding("mat")]



Outputs:

[[contextualized_embedding_layer_2("The"), contextualized_embedding_layer_2("cat"), contextualized_embedding_layer_2("sat"), contextualized_embedding_layer_2("on"), contextualized_embedding_layer_2("the"), contextualized_embedding_layer_2("mat")]]

8

OUTPUT OF ENCODER:

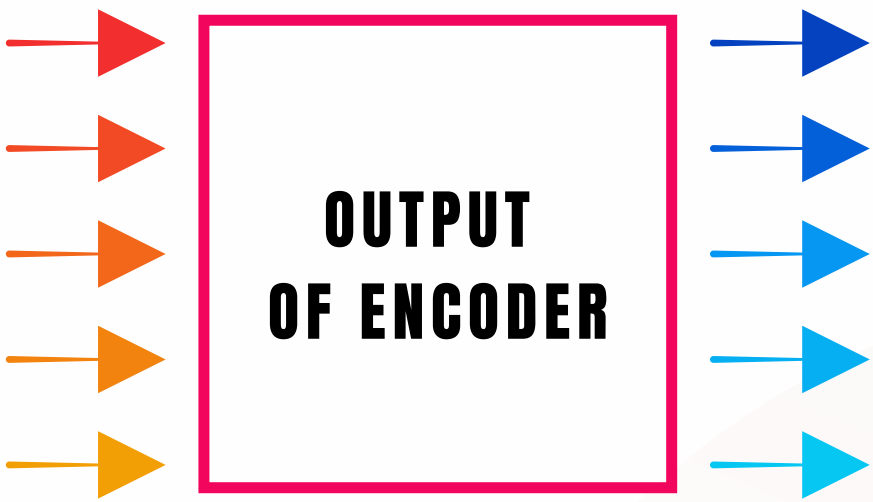
- The final output of the encoder is a sequence of context-aware representations for each token in the input sequence.
- These representations contain rich information about the input sequence and are passed on to the decoder for further processing in tasks like language translation or text generation.

STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

8

OUTPUT OF ENCODER:

Inputs:
Contextualized representations of tokens from the last encoder layer..



Outputs:
Sequence of context-aware representations for each token in the input sequence, containing rich information about the input sequence and ready to be passed to the decoder for further processing.

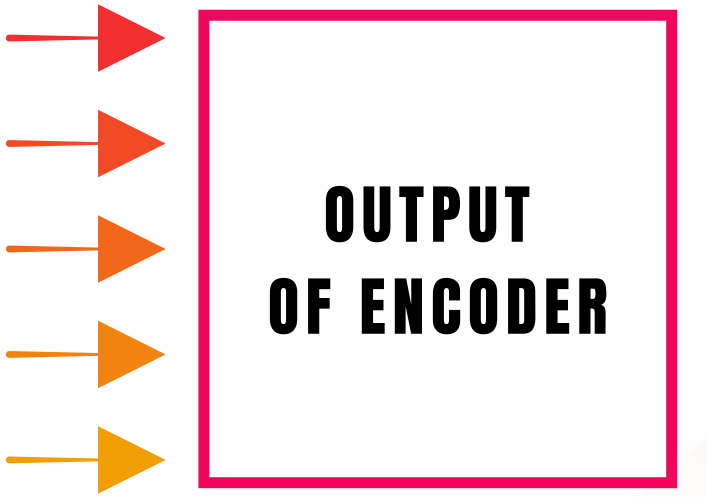
STEP-BY-STEP EXPLANATION OF ENCODER IN TRANSFORMER ARCHITECTURE

8

OUTPUT OF ENCODER:

Inputs:

[contextualized_embedding_layer_2("The"), contextualized_embedding_layer_2("cat"), contextualized_embedding_layer_2("sat"), contextualized_embedding_layer_2("on"), contextualized_embedding_layer_2("the"), contextualized_embedding_layer_2("mat")]



Outputs:

[final_contextualized_embedding("The"), final_contextualized_embedding("cat"), final_contextualized_embedding("sat"), final_contextualized_embedding("on"), final_contextualized_embedding("the"), final_contextualized_embedding("mat")]

Thank You

SPECIAL THANKS TO CHATGPT, OPEN AI, COPILOT, GEMINI FOR THE SUPPORT ON CONTENT

DINESHLAL