



Natural Language Processing (NLP)

Transformers 3

Residual Connections and Layer
Normalization

By:

Dr. Zohair Ahmed



www.youtube.com/@ZohairAI

Subscribe



www.begindiscovery.com

What are Transformers?

- Transformers are essentially a 14-step process for sequence modeling.
- Key for solving complex problems in natural language processing (NLP).
- Defined by an encoder-decoder architecture for tasks like language translation.



Prerequisites for Understanding Transformers

- Encoder-decoder architecture and their role in NLP.
- Concepts: words, tokens, embeddings, positional encoding.
- Basics of attention and its types.
- Layer normalization and residual connections.
- Feedforward neural networks, cross attention, and output generation.



High-Level Structure: Encoder and Decoder

- Transformer = Encoder-Decoder model.
- Example use case: Translating an English sentence (“How are you?”) into another language.
- The encoder processes input to a context vector; the decoder generates output from this context.



Breaking Down the Transformer: 14 Steps

- The internal workings of encoder and decoder can be divided into a clear 14-step sequence.



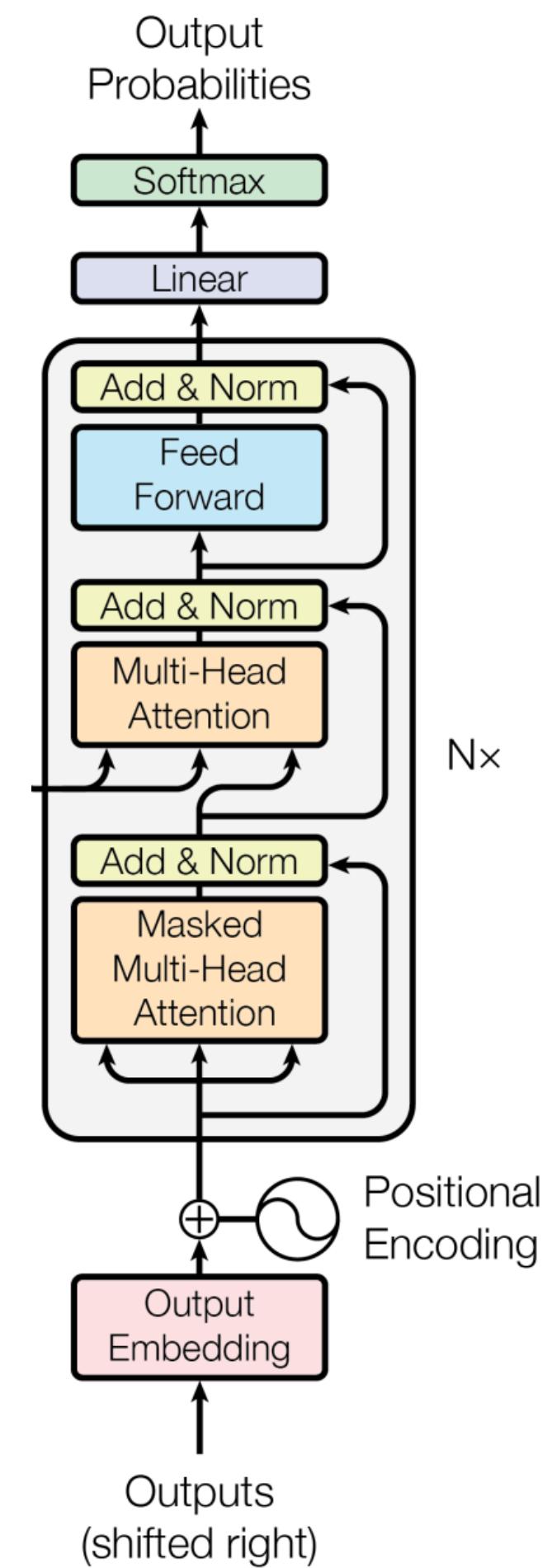
Encoder Steps (Steps 0-6)

- Step 0: Tokenization – split input into tokens (“How”, “are”, “you”).
- Step 1: Embedding – represent tokens as vectors.
- Step 2: Positional Encoding – add sequence information to embeddings.
- Step 3: Multi-head Attention – relate different parts of input.
- Step 4: Add & Norm – residual connection plus layer normalization.
- Step 5: Feedforward Layer – deeper processing.
- Step 6: Add & Norm (again).
- These six steps are repeated across six encoders (stacked architecture).



Decoder Steps (Steps 7-13)

- Step 7: Shifted Input + Embedding + Positional Encoding – process output from previous time step with start-of-sequence token.
- Step 8: Masked Multi-head Attention – allows only valid positions.
- Step 9: Add & Norm.
- Step 10: Cross Attention – queries from decoder, keys and values from encoder.
- Step 11: Add & Norm.
- Step 12: Feedforward Layer.
- Step 13: Add & Norm.
- These eight steps are repeated across six decoders.



Tensor2Tensor

- The code we used to train and evaluate our models is available at
- <https://github.com/tensorflow/tensor2tensor>
- The **Tensor2Tensor** repository includes a wide range of models and tools not just the Transformer.
- It was designed as a general-purpose library for training deep learning models across various tasks, including translation, image classification, and more.
- **Here's how you can locate it:**
- Look for the file: **transformer.py** inside the **tensor2tensor/models** directory.
- The configuration and training scripts are often found in **tensor2tensor/bin** or **tensor2tensor/utils**.
- You can also check the problem definitions in **tensor2tensor/data_generators** to see how datasets are prepared.



<bos> and <eos>

- **1. Encoder Input**
 - **Standard/Recommended:**
 - **Do NOT add <eos>** to the encoder input. Most implementations and the original Transformer paper do not use <eos> for the encoder input.
 - **Exception:**
 - Some toolkits or scripts may optionally add <eos> at the end of the encoder input for clear sequence boundary marking, but this is not required for the model to work.
 - **Rule:**
 - **You CAN add <eos> in encoder input if (and only if) your codebase/toolkit/dataset format needs it.**
 - For “Attention Is All You Need” and most frameworks: **omit <eos> on the encoder.**
- **2. Decoder Input (X)**
 - **NEVER add <eos> here.**
 - Always **prepend <bos>** at the start.
 - **3. Decoder Target/Output (Y)**
 - **ALWAYS append <eos>** at the end. The model must learn to predict <eos> as the last token as a stop signal.



In summary:

Step	Encoder Input Token	Encoder Output (Vector)	Decoder Input Token	Decoder Output (Target Token)
1	I	ctx_1	<bos>	مجھے
2	like	ctx_2	مجھے	بلیاں
3	cats	ctx_3	بلیاں	پسند
4	—	—	پسند	ہیں
5	—	—	ہیں	<eos>

Encoder Input	Decoder Input	Decoder Target (Y, what you want to generate/predict)
I like cats	<bos> مجھے بلیاں پسند ہیں	مجھے بلیاں پسند ہیں <eos>