NLP Transformers

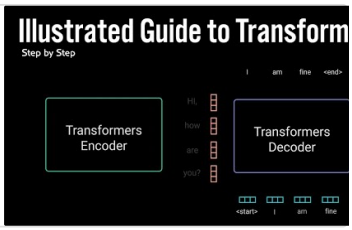
Transformers

Tutorial:

Illustrated Guide to Transformers Neural Network: A step by step explanation

Transformers are the rage nowadays, but how do they work? This video demystifies the novel neural network architecture with step by step explanation and illu...

https://www.youtube.com/watch?v=4Bdc55j80l8



1. Positional Encoding
    - store information about the order in data itself (not in the structure of the network)
    - while training it learns how to interpret those positional encodings
    - neural network learns the importance of word order from the data

2. Attention Mechanism
    - able to use the entire context of the story while generating the text
    - neural network structure that allows a text model to look at every single word in the original sentence when making a decision about how to translate a word in the output sentence
    - it learns to weight the relationship of each item in the input sequence to items in the output sequence

3. Self-attention
    - allows n-n to understand a word in the context of the words around it
    - it learns to weight the relationship of each item in the input sequence to all other items in the input sequence

ENCODER-DECODER ARCHITECTURE

Encoder

- maps an input sequence into an abstract, continuous representation that holds all the learned information of that input
- the encoder output is a continuous vector representation of the inputs

Decoder

- takes this continuous representation and step by step generates a single output while also being fed to the previous output
- feed previous outputs into the decoder recurrently until an “end of sentence” token, <end> is generated

DETAILED ARCHITECTURE

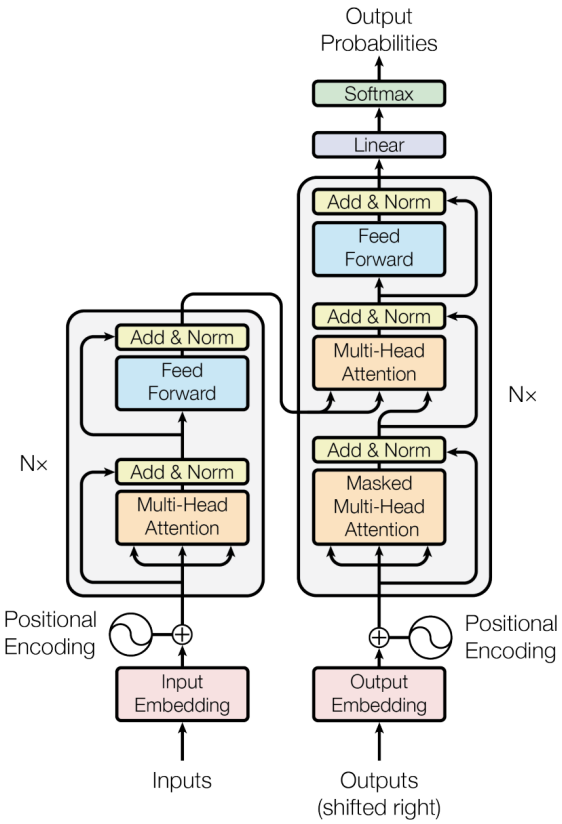
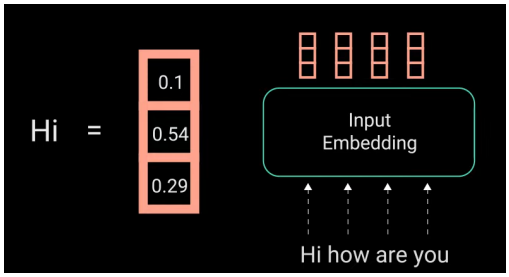


Figure 1: The Transformer - model architecture.

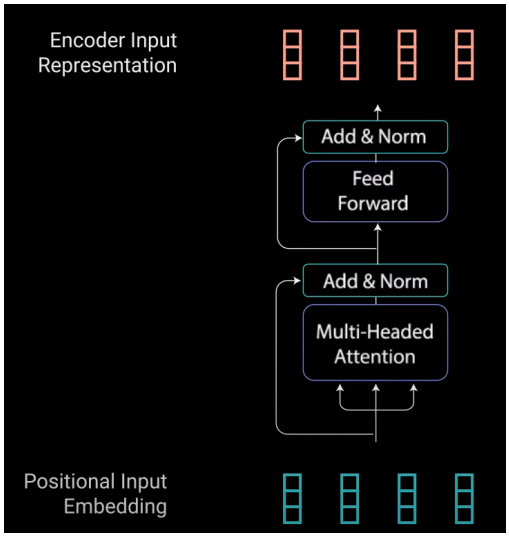
1. **Input Embedding** - it maps input text into the vectors of continuous values to represent that word



2. Positional Encoding - information about the positions is added to the input embedding (by *cos* and *sin* functions)

Encoder Layer

~ maps an input sequence into an abstract, continuous representation that holds all the learned information for that entire sequence

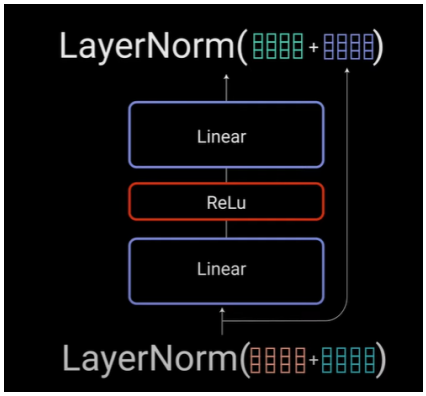


3. Multi-Headed Attention - applies the self-attention mechanism

   3.1 Self-Attention

   - associate each individual word in the input with the other words in that input
   - 3 distinct, fully connected layers: *query, key,* and *value* vectors

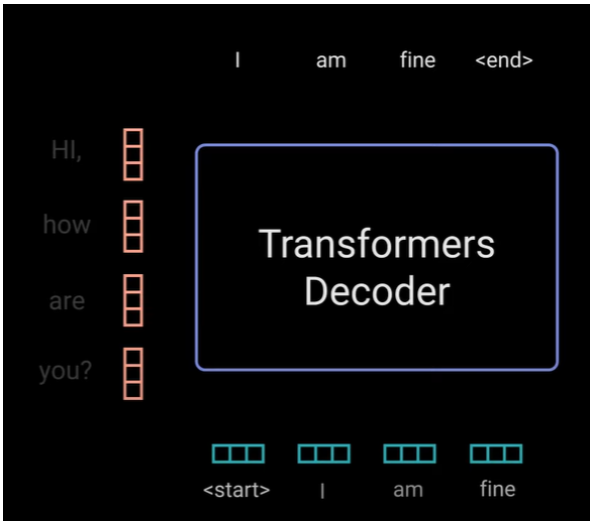
4. Residual Connection, Layer Normalization (stabilize the network) & Point-wise Feed Forward



Decoder Layer

~ generate text sequences

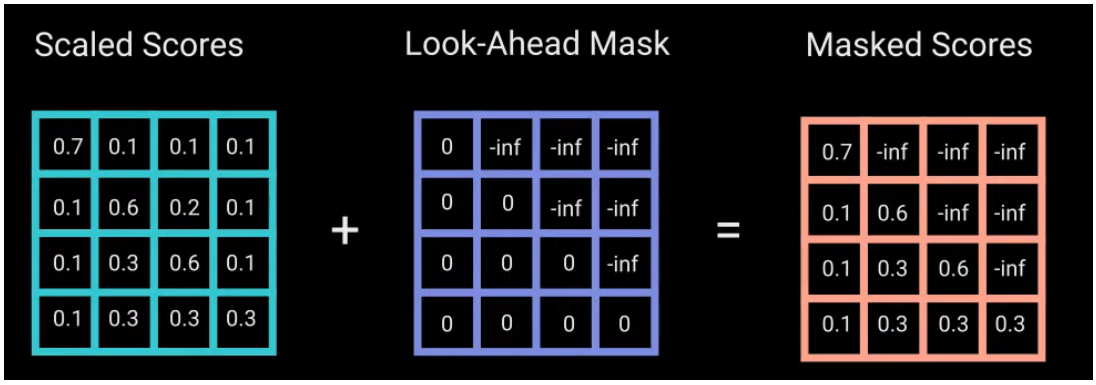
~ takes the list of previous outputs as inputs + encoder outputs that contain the attention information from the input



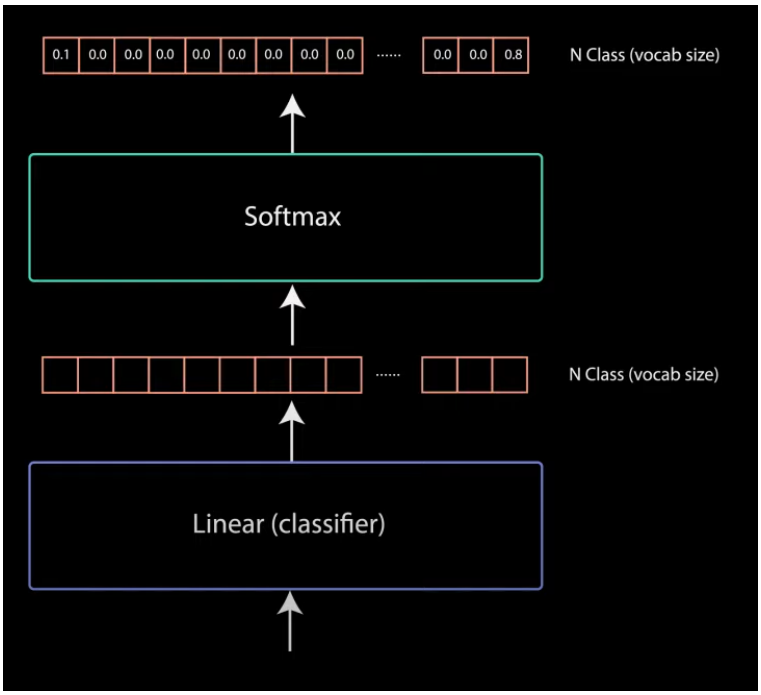
5. Output Embedding & Positional Encoding
6. Decoder Multi-Headed Attention 1 - computes the attention scores for the decoder's input

   6.1 Mask

   - prevent the decoder from looking at future tokens



7. Linear Classifier
8. Softmax



GPT-2

~ *Generative Pre-Training Transformer*

- to guess the next word in sentences

Fine-tuning

~a way of applying or utilizing transfer learning

- takes a model that’s already been trained for a given task and then tuning or tweaking this model to make it perform a second similar task