Macallan Ringstad 3/12/2024

What I have learned so far in Researching for Fin-GPT Transformer Predictions

I have done most of my research by reading different articles on Transformer architecture and then the different variables the transformer uses.

The first article was "The Ultimate Guide to Deep Learning" by Turing, this article gave a brief over view about what a transformer model is, the general architecture, what a transformer neural network is, a little about it's functioning, and some other information about transformer models in general. I'll list out some of the things I learned:

General Function:

- A Transformer neural network takes an input sentence and encodes it into two different sequences a word vector embedding sequence and a positional encoder sequence
- The word vectors are numerical representations of text
- The positional encoder sequence represent the words position in text as a vector
- The transformer combines these two vectors and sends it to Encoders and Decoders simultaneously, which is different from other LLM's

Architecture:

- The transformer model uses multi-head Encoder and Decoders to allow for parallel processing unlike recurring neural networks.
- The Encoder translates each word into a sequence of vectors, regardless of order, into a matrix of numbers, using a mathematical formula I do not understand as of now
- The Encoder does this multiple times and then normalizes these numbers as well
- The Decoder now instead of looking at all of the words in the sequence looks only at the words right behind it
- It then produces probabilities from the encoders and chooses what word to output based on that

Parameters:

- There is the Query which is what the model gives attention to in the sequence
- The Keys are tied to input elements to tell us what the element is offering and when it can get into the operation
- Values are put into a vector so we can get an average of these values to help with computation
- The Score Function, which I'm still trying to understand helps to determine which elements need more attention, the query and key are input into the score function and outputs key-query pairs/attention weight

The next article I read was "All you need to know about 'Attention' and 'Transformers' — In-depth Understanding — Part 1", I used this article to gain a greater understanding of how the Querys, Keys, and values flow through the model. I will summarize what I learned from this article,

- Instead of the score function, the query and keys, which are both multiplied with a key matrix which is a square matrix of the same dimension of the query and key matrix.
- Then a dot product is done between the new query and key vectors to, and then it is repeatedly normalized until it gets the value matrix.
- Then we are getting all of the value matrices and averaging them.

This is pretty much how the attention mechanism works for the transformer models.

Other than that this article was very similar than the first and nothing else stood out at me.

So far, this is a summary of what I am trying to understand and I hope I can keep learning more as I work on this project.