Date: __/__/20

M T W T F S

Assignment # 4

Question # 1
Explain Concept of Self-attention of Transformer mode? How it differ ...? mathematical

Self - Attention :

Self - Attention is a key Component of the transformer model Which was introduced by Vaswani et al. in Paper 'Attention is All you Need'

It is differs from traditional attention mechanism in its ability to capture relationships between different positions in a sequence, allowing the model to focus on different parts of input sequence when producing output e.g Sequence - to - Sequence tasks like machine translation.

Traditional Vs Self Attention Mechanism :

Checked By:__________

M T W T F S Date: ___ / ___ / 20 ___

In traditional attention mechanism, a context vector is computed based on the weighted sum of input sequence elements, where weights are determined by similarity b/w current decoding position & each position in the input sequence.

CV (used to compute output at current decoding position)

In self-attention, input sequence transformed into three vectors.

Query (Q)
Key (K)
Value (V) } → compute attention scores

& output → weighted sum of value where weights are determined by attention scores

Mathmatically :

Give input sequence

$X = (x_1, x_2, \ldots x_n)$

we compute Query (Q), key (K), value (V) matrices

$$\left.\begin{aligned} Q &= XW_Q \\ K &= XW_K \\ V &= XW_V \end{aligned}\right\}$$ learnable weights matrices

Self- attention scores are computed using dot product of Query & key

$$\text{Attention}(Q, K) = \frac{QK^T}{\sqrt{d_K}}$$

$\downarrow$ dimension of Key vector

Scores are scaled & passed through softmax function

$$\text{Attention\_weights} = \text{softmax}\left(\frac{\text{Attention}(Q K)}{\sqrt{d_K}}\right)$$

final output is a weighted sum of values vectors

Output = Attention_weights V

Question # 02

Summarize architecture of BERT & discuss how it differs from ~~[illegible]~~ orginal Focus on theoretical & key innovations?

BERT :

(Bidi-rectional Encoder Representations From transformers) is a pre-trained language model introduced. by google It builds upon orginal transformer model but introduces key innovations to address the limitations of Uni- directional language models.

Architecture :

1) Bidirectional Context
2) Masked language model (MLM)
3) Pre - training tasks
4) Transformer Encoder layers.

M T W T F S

1) Bidirectional Context :-

Bert Processes input text bi-directionally. It Considers both left & right during training.

2) MLM :

Bert utilizes a masked language model objective during pre-training Randomly Selected words in the input sequence are masked. & model is trained to predict these masked words based on their Context.

3) Pre-training Tasks :

Bert is pre-trained on 2 Unsupervised tasks

- MLM
- NSP (task involves predicting whether 2 sentences follow each other in training data.

∴ These help Bert Capture deeper Understanding of language semantics.

Checked By:____________

M T W T F S Date: __/__/20__

Key Difference From Original Transformer:

* Bi-directional Processing
* Task Specific Pre-training
* No Sequence to Sequence task.

~~Bi-directional (Bert introduces~~

Bi-directional Processing, capturing contextual info from both directions Original transformer which processes input sequentially.

Task-Specific Pre training: Bert pre-trains on Specific task like MLM & NSP, enable model to Capture broader Understanding of language Semantics & relationships.

No Sts: Unlike original transformer, Bert is not designed for Sequence-to-Sequence tasks.

~~[illegible]~~ ~~[illegible]~~

Q#03 :-

Consider a Scenerio where transformer model is struggling with long-range dependencies in input sequence ....
address this issue & explain why your approach could be effective.

To address the issue of long-range dependencies in a transformer model, one effective modification is to incorporate additional positional information in the form of 'Relative Positional encoding'

Modification Proposel :
R P E

In the original transformer model, Positional encoding are added to the input embeddings to Provide information about the positions of each tokens in the sequence.

Checked By:

M T W T F S

Date: __/__/20__

However, these absolute positional encoding may not effectively capture the relative distances b/w tokens, especially in long sequences.

The proposed modification involves introducing relative positional embeddings that explicitly model the relative distande b/w tokens.
This can be achieved by incorporating trainable parameters for encoding relative positions,
allowing model to learn the importance of different positional relationships during training.

Why it Could be Effective:

+ Improved capture of Large R Dependencies
+ Adaptability to sequence lengths
+ Enhanced training Dynamics.

1) Relative Positional encodings enable the model to better capture the relationships between tokens at varying distances

This added information helps model understand the context more effectively, especially in situations where long-range dependencies are crucial.

2) Unlike absolute positional encodings, which are fixed regardless of sequence length, relative positional encodings adapt dynamically to different sequence length.

3) The introduction of trainable parameters for relative positional encoding introduces additional flexibility during training.