Attention

- Intro:-
  - Helps LLMs understand & generate tent by focusing on the relevant parts of input sequences.

  - Attention allous the weigh the impor of diff words in a sentence, enabling better content understanding.

- Embeddings Recap:-

  - Embedding converts words into numerical vectors in a high-dim space.

  - Similar words are mapped to close points in this space.

  - "Apple" might map close to "Orange" in the content of fruits but close to "Phone" in a technology content.

- Attention Mechanism Basics:-

  - Self-Attention :- Each word in a sentence is compared w/ every other word to compute a relevance score.

- Key Components:-
  - Query (Q) :- Represents the word we're focusing on.
  - Key (K) :- Represent all words ~~we're~~ ~~focusing~~ in the sentence.
  - Value (V) :- The actual content of the words that are weighted.

- Mathematical Formulation:-

  - Dot Product:-

$$\text{Similarity} = Q \cdot K$$

if vector have unit length, dot product & cosine similarity are equivalent.

  - Cosine Similarity:-

$$\frac{Q \cdot K}{\|Q\| \times \|K\|}$$

  - Scaled Dot-Product

$$\frac{Q \cdot K}{\sqrt{d_k}}$$

- Used to prevent large dot products when vectors are high-dim.

- Softmax Function:-

  - Converts raw similarity scores into probabilites.

$$Softmax(x_i) = \frac{e^{x_i}}{\Sigma_j e^{x_j}}$$

Ensures That scores sum to 1 are positive.

- Attention Score Calculation:-

Steps:-

- Compute similarity scores using dot product or scaled dot product.

- Apply the softmax function to these scores.

- Multiply the result by the value vectors to get the final weighted content vectors

- Multi-Head Attention:-

  - Uses multiple sets of Q, K, V matrices to capture different aspects of word relationship.

- Process:-
  - Each "head" perform attention independently
  - Results are concatenated & linearly transforened to produce the final output.

- Advantages:-

  Allows the model to focus on different parts of The sentence simultanously.

- Value Matrix (V):-

Role:- · Transform the content vector into a form that is optimal for the next word prediction.

Interaction:-

-> Keys & Quries are used to calculate attention weights.

-> Values are transformed using these weights to get the final representation

Summary of the Attention process:-

-> Compute Similarities:- Between words using Q & K

-> Softmax Normalization:- Convert similarities into Normalization.

-> Weighted Sum:- Use weights to compute a weighted sum of the value vectors.

-> Output:- The resulting vector is used for further processing (e.g., predicting the next words).

- Application in Transformers:-

Attention is applied multiple times within a Transformer to iteratively refine the understating of content.

Tranformer Architecture:-

- Embedding -> Positional Encoding -> Multi-Head Attention -> Feedforward Network.

- Multi-Head attention is required accross layers to build a deep understanding of text.