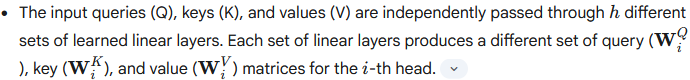
**Multi-head attention**

======================================================================

The multi-head attention mechanism in Transformers is an extension of the scaled dot product attention. Instead of performing a single attention calculation, it runs through the attention mechanism **multiple times in parallel** with different learned linear projections of the queries, keys, and values.

Here's a breakdown of how it works:

1. Multiple Linear Projections (Heads):



* These linear transformations project the original Q, K, and V into different representation subspaces. The idea is that each "head" can learn different aspects of the relationships between the words in the input sequence. For example, one head might focus on grammatical relationships, while another might focus on semantic relationships.

2. Scaled Dot Product Attention for Each Head:



* This results in h different output matrices, each representing the attention output from a different perspective or subspace.

3. Concatenation of Heads:

* The output matrices from all h attention heads are then concatenated along their last dimension (the feature dimension). If each head produces an output of dimension dv​, and there are h heads, the concatenated output will have a dimension of ‘h X dv’​.

4. Final Linear Transformation:

* The concatenated output is then passed through a final learned linear transformation (WO) to produce the final output of the multi-head attention layer. This linear layer projects the concatenated features back into the desired output dimension.

-------------------------------------------------------------------------------------------------------------

**Why Multi-Head Attention?**

**Captures diverse relationships:** By having multiple attention heads, the model can simultaneously attend to different parts of the input sequence and learn different types of relationships between them. A single attention head might struggle to capture all the nuances.

**Enriches representation:** Each head provides a different "view" or perspective on the input sequence. Concatenating these views allows the model to build a richer and more informative representation of each token, considering its relationships with other tokens from multiple angles.

**Improved performance:** Empirically, multi-head attention has been shown to significantly improve the performance of Transformer models on various tasks, especially in natural language processing.

-------------------------------------------------------------------------------------------------------------

In summary, multi-head attention allows the Transformer to effectively learn complex dependencies within a sequence by attending to information from different representation subspaces in parallel. It's a key innovation that contributes significantly to the power and versatility of the Transformer architecture.