Multi-Head Attention

The multi-head attention mechanism is a key component of the Transformer model. It improves upon traditional attention mechanisms by enabling the model to focus on multiple aspects of the input sequence simultaneously. Here's a breakdown of the components and their functions:

**Scaled Dot-Product Attention**

**Inputs**: Queries (Q), Keys (K), and Values (V).

**Process**:

**Score Calculation**: Compute attention scores by taking the dot product of queries and keys. The scores indicate how much focus a given query should place on each key.

**Scaling**: The scores are scaled down by dividing by the square root of the dimension of the keys (d\_k) to stabilize gradients and improve training.

**Softmax**: Apply a softmax function to convert scores into a probability distribution, represent

**Weighted Sum**: Multiply the weights by the values to get the output, which is a weighted sum of the values.

**Multi-Head Attention Function**

**Purpose:**  
The multi\_head\_attention function implements the multi-head attention mechanism, which enhances the model's ability to capture different aspects of the input sequence by attending to multiple parts of the sequence simultaneously. This process involves reshaping tensors for multi-head attention, applying attention mechanisms, and then combining the results.

**Process:**

**Linear Projections:**  
The function starts by projecting the input tensors—queries, keys, and values—into different subspaces using learned dense layers:

* **W\_q**: A dense layer that projects the input queries into a space of dimension d\_k \* h.
* **W\_k**: A dense layer that projects the input keys into a space of dimension d\_k \* h.
* **W\_v**: A dense layer that projects the input values into a space of dimension d\_v \* h.

These projections are crucial for the subsequent attention calculations, allowing the model to learn different aspects of the input data through separate linear transformations.

**Reshape for Multi-Head Attention:**  
After the linear projections, the function reshapes the projected tensors for multi-head attention:

* + The reshape\_for\_multihead function is used to transform the projected queries, keys, and values into a format suitable for attention across multiple heads. This reshaping divides the tensor into separate heads, each with its own representation of the input data.

**Scaled Dot-Product Attention:**  
The reshaped tensors are then fed into the scaled\_dot\_product\_attention function:

* + This function computes the attention scores by taking the dot product of the queries and keys, scaling the scores, and applying a softmax function to obtain attention weights.
  + These attention weights are then used to compute a weighted sum of the values, resulting in the attention output for each head.

**Revert Reshape:**  
Once the attention outputs are computed for each head, the function uses revert\_reshape to combine these outputs:

* + This function reshapes the attention outputs back into a format where all the heads are combined into a single tensor, consolidating the diverse information captured by each attention head.

**Final Linear Transformation:**  
Finally, a dense layer W\_o is applied to the combined output:

* + This layer projects the concatenated output of all attention heads back into the original dimensionality of the model, producing the final attention output tensor.