

1 Introduction

2 Tasks

2.1 Week 1

2.2 Rooshan Khan: Attention Method in bert.py

In this method we had to implement Attention method from class BertSelfAttention. The attention mechanism is given by:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

where Q , K , and V represent the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors.

I used method `torch.matmul` to multiply Q and transpose of K . I multiplied the result with `attention_mask` to apply the mask. The dimensions of `attention_mask` are `[bs, 1, 1, seq_len]`. The attention mask distinguishes between non-padding tokens and padding tokens. The non-padding tokens have a value of 0 while padding tokens have a value of a large negative number. The dimensions of `key_layer`, `query_layer` and `value_layer` are `[bs, num_attention_heads, seq_len, attention_head_size]`

Now I will tell how I concatenated all heads. When we transpose the tensor we change the shape of the tensor from `[bs, num_attention_heads, seq_len, attention_head_size]` to `[bs, seq_len, num_attention_heads, attention_head_size]`. This enables us to reshape the tensor to `[bs, seq_len, num_attention_heads * attention_head_size]`. After applying the transpose method the data sequence does not follow a contiguous order so we need to use **contiguous** method before using the **view** method.