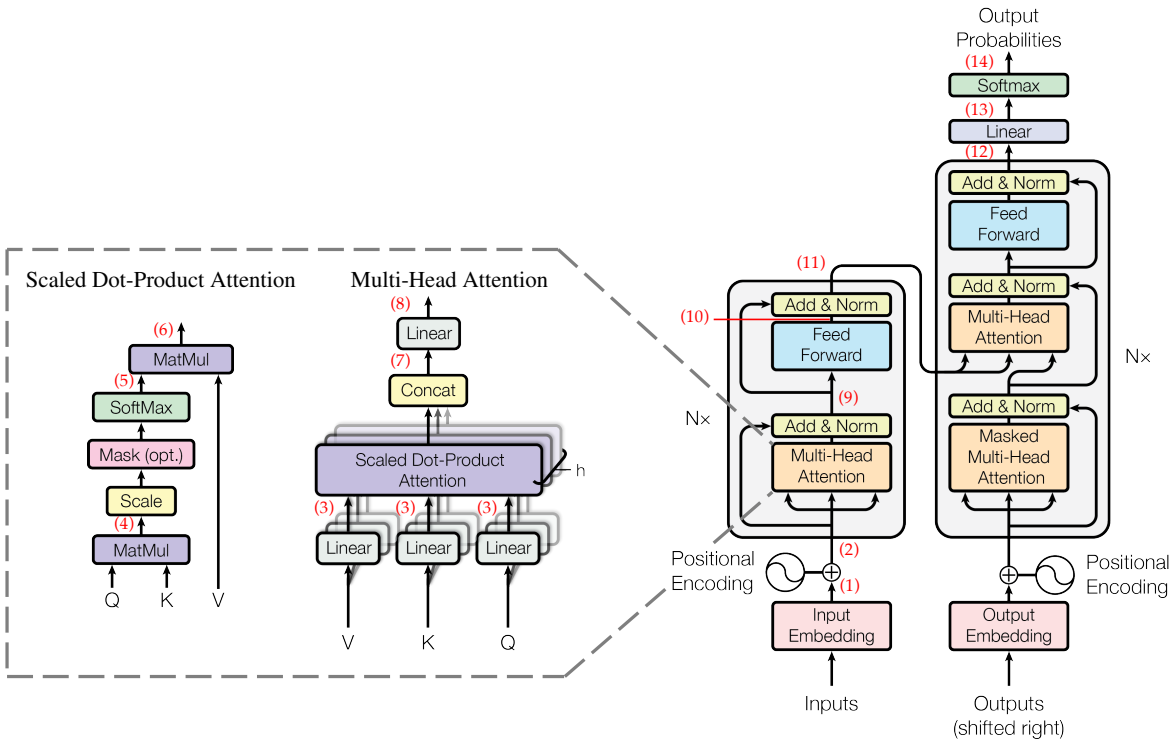
Week 4 - Pen & Paper Exercise

UvA - Deep Learning I

November 24, 2025

1 Knowing the Shapes is All You Need

Consider the following Transformer-based encoder-decoder architecture presented in Vaswani et al. (2017):



Let's say we use this architecture for machine translation. We consider the input sequence `She is very fast`, which we tokenize as follows:

Token text	`She`	`is`	`very`	`fast`	`<eos>`
Token ID	13864	4751	1345	34147	2

Therefore, the input of our encoder is the vector `[13864, 4751, 1345, 34147, 2]`. As it is a 1D vector with 5 components, we will represent its dimensionality as [5]. Assuming a context length of 512, the input vector would have dimensionality [512]. In practice, we would process multiple sequences in a batch, but for simplicity we will ignore the batch dimension in this exercise.

The first step is computing the embeddings for the input tokens. Considering an embedding size of 768, the dimensionality at (1) is [512, 768]. In the following questions, you will determine how the internal representation of the sequence changes after each layer of the encoder:

a) What is the dimension after adding the positional embeddings (2)?

[512, 768].

b) Now the input goes through the Multi-Head Attention block. The data first passes through three parallel linear layers, resulting in the Query (Q), Key (K) and Value (V) matrices that will be fed to each attention head. What is the shape of each of these variables (3)?
Hint: You can interpret the Linear Layers in the Multi-Head attention blocks as one linear layer that preserves the embedding dimension, or h linear layers that reduce the embedding dimension by a factor of h). Also, think whether some dimensions must be swapped for future computations.

Each head has size $768/8 = 96$, so the tensor is reshaped to [512, 8, 96]. To align dimensions for matrix multiplication, we transpose the sequence length and the number of heads, obtaining shapes [8, 512, 96] for Q, K and V.

c) What is the shape after multiplying queries by keys (4)? *Note: Keep the head dimension.*

[8, 512, 512], since the embedding dimension disappears after the multiplication.

d) What is the shape after scaling, masking and applying the softmax (5)?

Still [8, 512, 512], since none of these operations change the shape.

e) Finally, we multiply the attention weights by the values. What is the resulting shape (6)? And after concatenating the results from all heads (7)?

[8, 512, 96], and [512, 768] after concatenation.

f) We now apply a linear layer that preserves the input size. What is the shape at (8)?

[512, 768].

g) After multi-head attention, we apply the Add + Normalize step, where the tensors before and after the attention layer are added and normalized. What is the shape after this step (9)?

[512, 768].

h) The feed-forward network typically consists of two consecutive linear layers: one that expands the dimensionality and another that reduces it back to its original size. After that, there is another Add + Normalize layer. What are the dimensionalities after each of these two steps (10) and (11)?

[512, 768] and [512, 768].

The steps b–h are repeated for each of the N encoder blocks, resulting in the final encoder output.

Let's now move on to the decoder. Suppose we want to translate the input sequence into Spanish. Consider the following partial output sequence:

Token text	<bos>	Ella	es	muy
Token ID	12	321	2893	31217

The decoder now needs to predict a probability distribution for the next token.

j) What is the shape after all decoder blocks (12)? (Assume a context length of 512 for the decoder as well).

[512, 768].

k) The final linear layer transforms the embedding dimension into the vocabulary size, after which Softmax is applied to obtain a probability distribution. Assuming a vocabulary size of 32834, what are the shapes after the linear layer and the softmax (13), (14)?

[512, 32834].

Follow-up questions:

l) Is it required that the encoder and the decoder share the same context length?

No. Note that the only layer where different context lengths could be problematic is the Multi-Head attention block, where K and V are computed from the encoder output ($[L_{\text{enc}}, L_{\text{emb}}]$) and Q is computed from the internal representation of the decoder input ($[L_{\text{dec}}, L_{\text{emb}}]$). The first matrix multiplication QK^T results in shape $[L_{\text{dec}}, L_{\text{enc}}]$, which, when multiplying by V ($[L_{\text{enc}}, \frac{L_{\text{emb}}}{h}]$), results in shape $[L_{\text{dec}}, \frac{L_{\text{emb}}}{h}]$, so all operations are allowed.

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.