

This week we will work through a set of exercises on attention tensor shapes, the structure of multi-head attention, hand-computed single-head and multi-head forward passes, and the block-matrix view used in practical MHA implementations.

1. Single-Head Attention Shapes

The figure below shows a vanilla single-head attention block. Assume the input tensor has shape $X \in \mathbb{R}^{B \times n \times d}$, where B is the batch size, n is the sequence length, and d is the token embedding dimensionality. Assume further that the dimension of the key vector d_k and dimension of the value vector d_v for the attention are both equal to d . This question asks you to determine the shapes of relevant tensors.

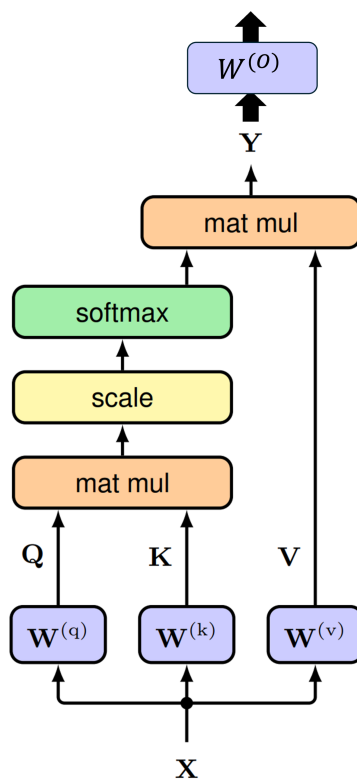


Figure 1: Single-head attention block.

- (a) Please provide the shapes for the query/key/value input projection weight matrices $W^{(q)}$, $W^{(k)}$, $W^{(v)}$.

Solution: $W^{(q)} \in \mathbb{R}^{d \times d}$, $W^{(k)} \in \mathbb{R}^{d \times d}$, $W^{(v)} \in \mathbb{R}^{d \times d}$ since each linear projection weight matrix maps from the input embedding space of dimension d to the query/key/value space of dimension d .

Although not commonly used in practice, we could also design the attention block to have arbitrary query-key/value dimensions. Generally, we could specify the dimension of the key vector d_k and the dimension of the value vector d_v independently. Note that since the query vectors need to do inner product with the key vectors, the dimension of the query vectors should also be d_k . In this case, the shapes of the projection weight matrices would be $W^{(q)} \in \mathbb{R}^{d \times d_k}$, $W^{(k)} \in \mathbb{R}^{d \times d_k}$, $W^{(v)} \in \mathbb{R}^{d \times d_v}$.

(b) Please provide the shapes for the query/key/value activations Q, K, V after the input projection.

Solution: $Q \in \mathbb{R}^{B \times n \times d}$, $K \in \mathbb{R}^{B \times n \times d}$, $V \in \mathbb{R}^{B \times n \times d}$ since $Q = XW^{(q)}$, $K = XW^{(k)}$, $V = XW^{(v)}$.

(c) Please provide the shapes for the result after the first matmul QK^\top , the scaled matmul result $\frac{QK^\top}{\sqrt{d}}$, and the softmax result (i.e. the attention weights) $\text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)$. Note that the formulae here are for unbatched inputs.

Solution: The shapes are $QK^\top \in \mathbb{R}^{B \times n \times n}$ since we apply the matmul formula independently for each element in the batch (i.e. we broadcast the matmul over the batch dimension), $\frac{QK^\top}{\sqrt{d}} \in \mathbb{R}^{B \times n \times n}$ has the same shape, and $\text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) \in \mathbb{R}^{B \times n \times n}$ again has the same shape.

(d) Please provide the shapes for the result of the second matmul Y , the output projection weight matrix $W^{(o)}$, and the final output of the attention block after the output projection $YW^{(o)}$.

Solution: $W^{(o)} \in \mathbb{R}^{d \times d}$ because the output projection maps from the pre-linear attention output (which has dimension d) back to the input embedding space (which also has dimension d).

The shapes are $Y = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V \in \mathbb{R}^{B \times n \times d}$ since we apply the matmul formula independently for each element in the batch, $W^{(o)} \in \mathbb{R}^{d \times d}$, and $YW^{(o)} \in \mathbb{R}^{B \times n \times d}$.

2. Multi-Head Attention Mechanism

The figure below shows a sketch of the multi-head attention mechanism. In this question we will review how we can have multiple attention heads from a single input, and how each head's outputs are combined to produce a single MHA output tensor.

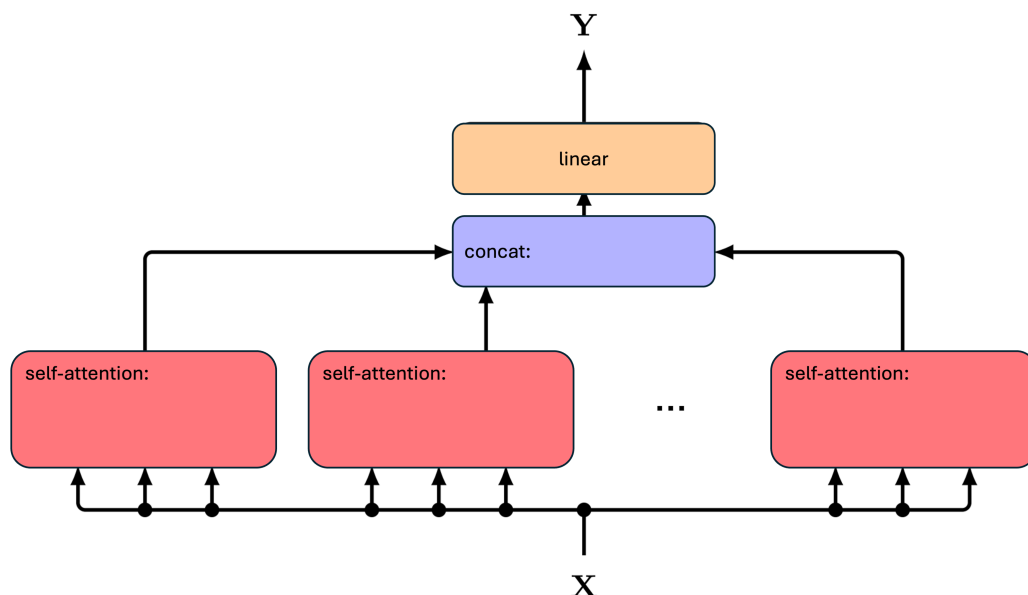


Figure 2: Multi-head attention block.

- (a) Please draw on the figure where each query/key/value input projection weight matrices $W_h^{(q)}$, $W_h^{(k)}$, $W_h^{(v)}$ are located.
- (b) Please draw on the figure where each query/key/value activation Q_h , K_h , V_h and the per-head attention output H_h are calculated.
- (c) Please draw on the figure how the per-head attention outputs $H_1, H_2, \dots, H_{n_{\text{heads}}}$ are combined to produce a single tensor before the final output projection linear layer.

Solution:

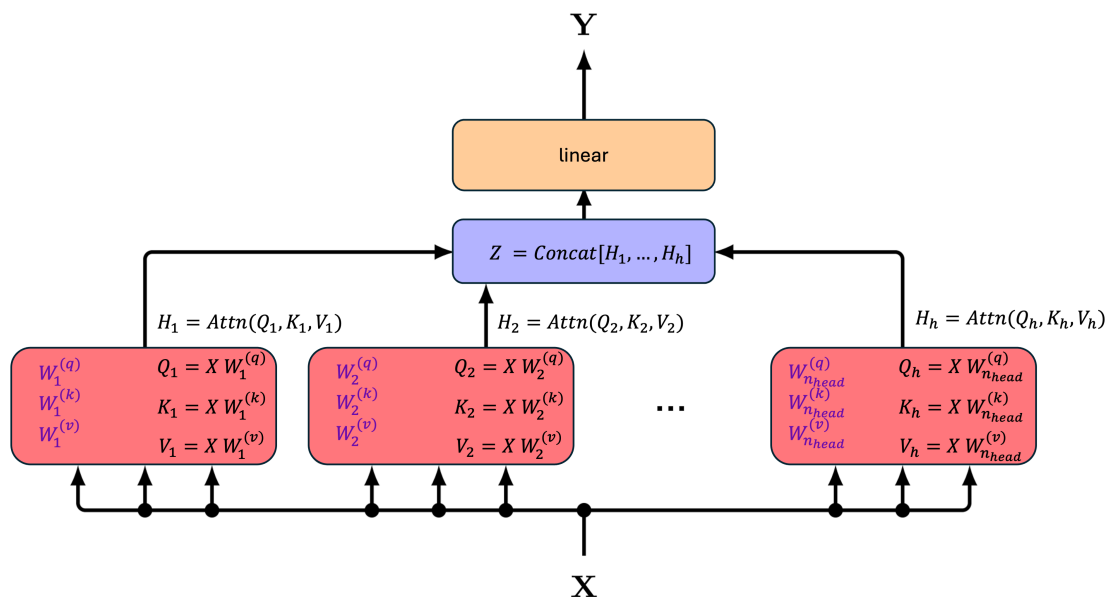


Figure 3: Multi-head attention block with solution annotations.

3. Single Head Attention Forward Pass

Recall the single head attention mechanism:

$$Q = XW^{(q)}$$

$$K = XW^{(k)}$$

$$V = XW^{(v)}$$

$$H = \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

$$Y = HW^{(o)}$$

In this question, we consider a toy example where sequence length $n = 2$, hidden dimension $d = 2$, and the input is not batched. Assume the input $X \in \mathbb{R}^{n \times d}$, the input projection weights $W^{(q)}$, $W^{(k)}$, $W^{(v)} \in \mathbb{R}^{d \times d}$, and the output projection weight $W^{(o)} \in \mathbb{R}^{d \times d}$ bear the following values:

$$X = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad W^{(q)} = \begin{bmatrix} 3 & -1 \\ 1 & 1 \end{bmatrix}, \quad W^{(k)} = \begin{bmatrix} 2 & -2 \\ 1 & 3 \end{bmatrix}, \quad W^{(v)} = \begin{bmatrix} 2 & -2 \\ 0 & 2 \end{bmatrix}, \quad W^{(o)} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

(a) Compute the matrices Q , K , and V .

Solution:

$$\begin{aligned} Q &= XW^{(q)} \\ &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 1 & 1 \end{bmatrix} \\ K &= XW^{(k)} \\ &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \\ V &= XW^{(v)} \\ &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \end{aligned}$$

(b) Given that the attention weights are approximately

$$\text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) \approx \begin{bmatrix} 1.00 & 0.00 \\ 0.50 & 0.50 \end{bmatrix}$$

Please compute the output Y of the single head attention.

Solution: We first compute the output of the attention head H :

$$\begin{aligned} H &= \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \\ &= \begin{bmatrix} 1.00 & 0.00 \\ 0.50 & 0.50 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \end{aligned}$$

Then we compute the final output Y :

$$\begin{aligned} Y &= HW^{(o)} \\ &= \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 2 & 1 \end{bmatrix} \end{aligned}$$

4. MHA with Block Matrices For multi-head attention with number of heads $n_{\text{heads}} > 1$, in practice we often do not explicitly keep separate projection matrices $W_h^{(q)}$, $W_h^{(k)}$, $W_h^{(v)}$ for each head in the forward computation. Instead, we form single large projection matrices $W^{(q)}$, $W^{(k)}$, $W^{(v)}$ and compute

$$\begin{aligned} Q &= XW^{(q)} \\ K &= XW^{(k)} \\ V &= XW^{(v)}. \end{aligned}$$

Then each head operates on the appropriate feature chunk of Q , K , and V .

For example, if $n_{\text{heads}} = 2$ and hidden dimension $d = 64$, then each head has size $d_h = 32$. Head 0 operates on $Q[:, : 32]$, $K[:, : 32]$, $V[:, : 32]$, and head 1 operates on $Q[:, 32 :]$, $K[:, 32 :]$, $V[:, 32 :]$.

To show why this is valid, recall the block-matrix identity (here the square brackets denote matrix concatenation instead of indexing)

$$A[B_0 \mid B_1] = [AB_0 \mid AB_1].$$

Here B_0 and B_1 are a split of B along its column axis.

(a) Consider the two-head case $n_{\text{heads}} = 2$ and assume $d = 2d_h$. Write

$$W^{(q)} = [W_0^{(q)} \mid W_1^{(q)}],$$

where $W_0^{(q)}, W_1^{(q)} \in \mathbb{R}^{d \times d_h}$. Show that computing $Q = XW^{(q)}$ and then splitting Q along the feature dimension is equivalent to separately computing $Q_0 = XW_0^{(q)}$ and $Q_1 = XW_1^{(q)}$.

Solution: Using the block-matrix identity,

$$\begin{aligned} Q &= XW^{(q)} \\ &= X[W_0^{(q)} \mid W_1^{(q)}] \\ &= [XW_0^{(q)} \mid XW_1^{(q)}]. \end{aligned}$$

If we define

$$Q_0 = XW_0^{(q)} \quad \text{and} \quad Q_1 = XW_1^{(q)},$$

then

$$Q = [Q_0 \mid Q_1].$$

Therefore, computing the full matrix product $XW^{(q)}$ and then splitting the output into two chunks is exactly the same as separately computing the per-head query matrices Q_0 and Q_1 .

(b) Explain why the same argument applies to K and V , and conclude why chunking Q , K , and V after the large projection is equivalent to explicitly keeping track of separate per-head projection matrices.

Solution: The same reasoning applies to the key and value projections:

$$\begin{aligned} W^{(k)} &= [W_0^{(k)} \mid W_1^{(k)}], & K &= XW^{(k)} = [XW_0^{(k)} \mid XW_1^{(k)}] = [K_0 \mid K_1], \\ W^{(v)} &= [W_0^{(v)} \mid W_1^{(v)}], & V &= XW^{(v)} = [XW_0^{(v)} \mid XW_1^{(v)}] = [V_0 \mid V_1]. \end{aligned}$$

So the first d_h columns of Q, K, V are exactly the tensors for head 0, and the next d_h columns are exactly the tensors for head 1. Therefore each head sees exactly the same Q_h, K_h, V_h that it would have seen if we had computed separate per-head projections from the start.

Hence, using a single large projection matrix followed by splitting into chunks is mathematically equivalent to explicitly keeping track of separate $W_h^{(q)}, W_h^{(k)}, W_h^{(v)}$ matrices for each head. The same argument extends immediately from 2 heads to n_{heads} heads by writing each projection matrix as a concatenation of n_{heads} column blocks.

Note: be careful about calculating the attention weights. In PyTorch, usually we could reshape Q, K, V to have shape $B \times n \times n_{\text{heads}} \times d_h$. For HW3, reading the torch batched matrix multiplication documentation or using einops can be helpful.

5. Multi-Head Attention Forward Pass

Recall the multi-head attention equations:

$$\begin{aligned} Q_h &= XW_h^{(q)} \\ K_h &= XW_h^{(k)} \\ V_h &= XW_h^{(v)} \\ H_h &= \text{Attention}(Q_h, K_h, V_h) \\ Y &= \text{Concat}[H_0, \dots, H_{n_{\text{heads}}-1}]W^{(o)}. \end{aligned}$$

Adopting the block matrix convention introduced in the previous question, we can group together

$$\begin{aligned} Q &= [Q_0 \mid Q_1 \mid \dots \mid Q_{n_{\text{heads}}-1}] = X[W_0^{(q)} \mid W_1^{(q)} \mid \dots \mid W_{n_{\text{heads}}-1}^{(q)}] = XW^{(q)}, \\ K &= [K_0 \mid K_1 \mid \dots \mid K_{n_{\text{heads}}-1}] = X[W_0^{(k)} \mid W_1^{(k)} \mid \dots \mid W_{n_{\text{heads}}-1}^{(k)}] = XW^{(k)}, \\ V &= [V_0 \mid V_1 \mid \dots \mid V_{n_{\text{heads}}-1}] = X[W_0^{(v)} \mid W_1^{(v)} \mid \dots \mid W_{n_{\text{heads}}-1}^{(v)}] = XW^{(v)}. \end{aligned}$$

In this question, we consider a toy example with $n_{\text{heads}} = 2$, sequence length $n = 2$, hidden dimension $d = 4$, and per-head hidden dimension $d_h = 2$. The input is not batched, so $X, Q, K, V \in \mathbb{R}^{2 \times 4}$.

(a) Suppose for a particular input X , the projected query, key, and value matrices are

$$Q = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}, \quad K = \begin{bmatrix} 2 & 0 & 1 & 1 \\ 1 & 3 & 0 & 2 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 3 & 1 & 0 & 2 \end{bmatrix}.$$

Compute the per-head query, key, and value matrices $Q_0, Q_1, K_0, K_1, V_0, V_1$ by splitting Q, K , and V along the feature dimension.

Solution: Since each head has feature dimension $d_h = 2$, head 0 uses the first two columns and head 1 uses the last two columns:

$$\begin{aligned} Q_0 &= Q[:, : 2] = \begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix}, & Q_1 &= Q[:, 2 :] = \begin{bmatrix} 3 & 4 \\ 7 & 8 \end{bmatrix}, \\ K_0 &= K[:, : 2] = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}, & K_1 &= K[:, 2 :] = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}, \\ V_0 &= V[:, : 2] = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}, & V_1 &= V[:, 2 :] = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}. \end{aligned}$$

(b) Suppose for another input X , the attention outputs for the two heads are

$$H_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

and suppose the output projection matrix is

$$W^{(o)} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Compute the concatenated matrix $H = \text{Concat}[H_0, H_1]$ and then compute the final output $Y = HW^{(o)}$.

Solution: We first concatenate the two heads along the feature dimension:

$$\begin{aligned} H &= \text{Concat}[H_0, H_1] \\ &= \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}. \end{aligned}$$

Then we multiply by the output projection matrix:

$$\begin{aligned} Y &= HW^{(o)} \\ &= \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 1 & 2 & 2 \\ 1 & 3 & 2 & 2 \end{bmatrix}. \end{aligned}$$